

SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Diajukan kepada Fakultas Teknik

Universitas Negeri Yogyakarta

Untuk Memenuhi Sebagian Persyaratan

Guna Memperoleh Gelar Sarjana Pendidikan Teknik Informatika



Oleh:

ANDHIKA LADY MAHARSI

08520241041

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK**

UNIVERSITAS NEGERI YOGYAKARTA

2013

PERSETUJUAN

Skripsi yang berjudul **“Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika”** telah disetujui pembimbing untuk diujikan.

Yogyakarta, 18 September 2013

Dosen Pembimbing,

A handwritten signature in black ink, appearing to read 'Eko Marpanaji', followed by a horizontal line.

Dr. Eko Marpanaji.

NIP. 19670608 199303 1 001

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini menyatakan dengan **sebenarnya** bahwa skripsi ini adalah hasil karya saya sendiri. Sepanjang pengetahuan saya tidak ada bagian dari skripsi ini yang merupakan hasil karya atau pendapat orang lain kecuali saya jadikan sebagai acuan dengan menyertakan sumber dengan mengikuti etika penulisan karya ilmiah yang seharusnya.

Yogyakarta,

Peneliti,



Andhika Lady Maharsi

PENGESAHAN

Skripsi yang berjudul "Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika" yang disusun oleh Andhika Lady Maharsi NIM. 08520241041 telah dipertahankan di depan Dewan Penguji pada tanggal 29 Oktober 2013 dan dinyatakan lulus.

Dewan Penguji

Nama	Jabatan	Tanda tangan	Tanggal
Dr. Eko Marpanaji	Ketua Penguji		28 Desember 2013
Handaru Jati, Ph.D	Sekretaris Penguji		27 November 2013
Slamet, M.Pd	Penguji Utama		26 November 2013

Yogyakarta, 7 Januari 2014

Fakultas Teknik UNY

Dekan,


Dr. Moch. Bruri Triyono

Nip. 19560216 198603 1 003

HALAMAN MOTTO

“Sesungguhnya apa yang tertulis di halaman ini dan halaman kata pengantar adalah sungguh fluktuatif tergantung ego dan suasana hati peneliti. Pernah hendak ditulis ‘Nikmat Tuhan mana yang kamu dustakan’ lalu berubah menjadi ‘Jalani saja, nikmati proses, selangkah demi selangkah, tahu-tahu kamu akan terkejut telah berjalan sejauh itu’ lalu berubah lagi menjadi ‘Jangan percayai setiap motto kecuali kamu atau orang lain sudah membuktikannya. Termasuk yang tertulis dalam halaman ini’ lalu peneliti berpikir motto yang terakhir itu adalah yang paling baik”

Peneliti.

www.jenganten.com

www.andhikalady.blogspot.com

PERSEMBAHAN

Untuk **Bapak Sukirno, S.Pd** dan **Ibu Surtiyah**.

Dua manusia hebat yang membuat saya ada dan eksis hingga sekarang.

Untuk adik-adik saya yang ganteng-ganteng:

Arie Varian Akbari dan **Abyan Faishal Mahib**,

Siapapun pembaca blog saya di

www.andhikalady.blogspot.com dan

www.jenganten.com

siapapun kamu, kehadiran dan kunjungan kamu sangat berarti di hati saya.

Untuk diri saya sendiri,

Untuk siapapun yang hendak menyelami keindahan Genetika dalam dunia *true*
and false.

SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA

Oleh:
Andhika Lady Maharsi
NIM. 08520241041

ABSTRAK

Penjadwalan kegiatan belajar mengajar dalam suatu sekolah adalah hal yang sangat kompleks. Hal ini tidaklah menjadi problem yang serius ketika sekolah tersebut memiliki jumlah kelas yang sedikit dengan kuantitas pertemuan belajar mengajar yang minimal. Namun hal ini akan menjadi kendala tersendiri ketika dihadapkan dengan banyaknya faktor *constraint* penentuan jadwal, misalnya kelas yang banyak, ruangan yang terbatas, dan jumlah guru yang terbatas. Permasalahan ini sering disebut dengan University Timetabling Problems (UTP). Contoh permasalahan yang kerap terjadi adalah kesulitan untuk menempatkan slot jadwal supaya tidak terjadi tabrakan, efisien waktu dengan *effort* yang minimal. Ada pula kemungkinan jumlah jam mengajar guru yang berlebih. Untuk itulah dibutuhkan adanya optimasi untuk merancang sistem penjadwalan dengan meminimalisir *error* jadwal sehingga kegiatan belajar dapat terlaksana dengan optimal.

Salah satu metode optimasi untuk permasalahan ini adalah melalui pendekatan Algoritma Genetika. Algoritma Genetika merupakan pendekatan komputasional yang terinspirasi dari teori genetika dan evolusi Mendell untuk menyelesaikan masalah yang membutuhkan optimasi.

Hasil pengaplikasian Algoritma Genetika sebagai pendekatan dalam optimasi penjadwalan mata pelajaran sekolah dihasilkan pencapaian nilai *fitness* yang optimal dengan rerata kurang dari 200 iterasi. Kemudian diuji dari faktor *correctness*, dihasilkan jumlah *error* KLOC sebanyak 0 line. Diuji dengan faktor *functionally* dihasilkan tidak ada satupun fungsi primer dan sekunder yang tidak berjalan dengan semestinya. Diuji dengan faktor *usability* dihasilkan kepuasan penggunaan sebesar 88,89 % dan termasuk dalam kategori usabilitas yang sangat baik. Diuji dengan faktor *portability* dengan mencoba dalam berbagai environmental yang berbeda dihasilkan aplikasi dapat berjalan dengan semestinya dalam semua lingkungan.

Kata kunci: Penjadwalan, Optimasi, Algoritma Genetika

KATA PENGANTAR

Assalamu Alaikum Wr Wb,

Segala Puji bagi Allah swt karena tanpa hidayah dan anugrah yang Dia berikan, saya tidak akan mampu menyelesaikan skripsi berjudul Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika yang merupakan prasyarat untuk mendapatkan gelar sarjana pendidikan di Universitas Negeri Yogyakarta.

Skripsi ini tidak dapat tersusun tanpa bantuan dari orang-orang hebat berikut ini:

1. Untuk Bapak saya, Sukirno, S.Pd. Karena bapak saya bisa lahir, karena beliau pula saya dapat menyelesaikan pendidikan sampai sekarang.
2. Untuk Ibu saya, Surtiyah. Karena doa Ibu pula saya dapat terus bertahan dan tabah dalam setiap jejak kehidupan.
3. Untuk adik-adik saya, Arie Varian Akbari dan Abyan Faishal Mahib. Kita adalah triple-A paling keren di dunia.
4. Untuk dosen pembimbing saya, Dr. Eko Marpanaji. Atas bimbingan dan dorongan semangat untuk menemukan solusi dari Algoritma Genetika.
5. Untuk Pak Arif, Pak Imam dan Bu Nanik dari SMA N 1 Kalasan. Karena telah membantu dan membimbing saya dalam perolehan data *usability*.
6. Untuk Mbak Maria Sekararum Winahyu. Karena kamu adalah orang yang mengajari saya untuk *bitchy* dengan *elehan*.

7. Untuk teman-teman les tari Bali. Ari, Ana, Yanti, Ajeng. Karena menari bersama kalian merupakan pelipur lara dan pembangkit semangat.
8. Untuk teman-teman Makeup Nista. Arum, Puput, Ephong, Momon, Tintas, Dila, Popon. Saya kangen berdandan bersama kalian.
9. Untuk Herjuno Satriadi. Karena saya teringat pada sebuah percakapan di masa lalu, kamu ingin supaya saya sukses dan berhasil lulus kuliah dengan hasil yang mengagumkan.
10. Untuk Mas Yoga Hanggara, Nurul, Yudan, Nyoman. Karena telah membantu proses uji *portability*.
11. Untuk kawan-kawan Kombre kelas E 2008 yang tak henti-hentinya saling dukung. Termasuk pula yang datang di ujian pendadaran saya, Yudan, Rosyi, Tia, Anggit, dan Faris.
12. Untuk tim Audiogenic Band, Beb Oki Wicaksono, Beb Tri Nurika Cahyadi, Beb Khoirul Anam, Beb Johari, Beb Yudha. Kebersamaan bermain band adalah momen tidak terlupakan.
13. Untuk tim ACS Gamatechno, Pak Awal, Mba Rosa, Mba Wati, Angger, Mba Dyah, Mba Tika, Monica, Ajeng, Mas Nyoman, Mas Galih dan kawan-kawan. Terima kasih atas kerjasama selama ini. Saya banyak belajar menjadi *application support* dan memiliki rekan tim yang hebat.
14. Untuk Arrizal Bayu, atas karena janji yang menyebalkan sehingga saya harus menulis ucapan terimakasih dan menampung namanya.
15. Untuk Mas Aji Prasetyo. Terimakasih atas dukungannya. *Let's the little dream keep burning.*

Ucapan terimakasih juga saya sampaikan untuk setiap pihak yang tidak dapat disebutkan satu per satu.

Peneliti menyadari bahwa skripsi ini masih jauh dari sempurna. Untuk itu, segala kritik, saran, dan masukan sangat peneliti butuhkan untuk pengembangan penelitian lebih lanjut lagi terutama yang berkaitan dengan Algoritma Genetika.

Yogyakarta, 2013

Peneliti,

Andhika Lady Maharsi

DAFTAR ISI

PERSETUJUAN	ii
SURAT PERNYATAAN	iii
PENGESAHAN	iv
HALAMAN MOTTO	v
PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN.....	xviii
BAB I. PENDAHULUAN.....	20
A. Latar Belakang	20
B. Identifikasi Masalah	22
C. Pembatasan Masalah	23
D. Rumusan Masalah	23
E. Tujuan Penelitian	23
F. Manfaat Penelitian	24

BAB II. LANDASAN TEORI	25
A. Algoritma Genetika.....	25
B. Java	29
C. Perancangan Perangkat Lunak.....	29
D. Kualitas Perangkat Lunak (<i>Software Quality</i>)	30
E. Kualitas <i>Correctness</i>	32
F. Kualitas <i>Functionally</i>	34
G. Kualitas <i>Usability</i>	35
H. Kualitas <i>Portability</i>	37
I. Penelitian yang Relevan.....	38
BAB III. METODOLOGI PENELITIAN	40
A. Model dan Metode Penelitian	40
B. Sumber Informan	41
C. Lokasi Penelitian.....	41
D. Teknik Pengumpulan Data.....	42
E. Teknik Analisis Data.....	60
BAB IV PEMBAHASAN PENELITIAN.....	65
A. Analisis Kebutuhan	65
B. Desain Penelitian	66

C. Perancangan Penelitian	67
D. Penerapan Sistem Algoritma Genetika terhadap Desain	85
E. Pengujian Sistem.....	91
BAB V. KESIMPULAN DAN SARAN.....	103
A. Kesimpulan	103
B. Saran	104
DAFTAR PUSTAKA.....	106
LAMPIRAN.....	108

DAFTAR TABEL

Tabel 1. Perkiraan Jumlah <i>Error</i> KLOC Teori McConell.....	33
Tabel 2. Perkiraan Jumlah <i>Error</i> KLOC Teori Pressman.....	33
Tabel 3. Kriteria Kelolosan dalam Aspek <i>Functionally</i>	35
Tabel 4. Istilah dalam Algoritma Genetika.....	43
Tabel 5. Contoh Tabel <i>Fitness</i> Cost.....	49
Tabel 6. Sintaks Java untuk Penentuan Probabilitas.....	49
Tabel 7. Contoh Sintaks Java untuk <i>Roulette Wheel</i>	50
Tabel 8. Contoh Individu Penjadwalan.....	51
Tabel 9. Tabel <i>Test case</i> untuk Uji <i>Functionally</i>	56
Tabel 10. Penjabaran Butir Pertanyaan Aspek <i>Usability</i>	57
Tabel 11. Kriteria Kelolosan Variabel <i>Functionally</i>	61
Tabel 12. Data Skor untuk Kualitas <i>Usability</i>	62
Tabel 13. Tabel Interval Nilai Kualitas <i>Usability</i>	63
Tabel 14. Kriteria Kelolosan Analisis <i>Portability</i>	64
Tabel 15. Contoh Skema Jadwal.....	66
Tabel 17. Contoh Tabel Jadwal.	68
Tabel 18. Contoh Penjadwalan Digambarkan dengan Kode Mapel	69
Tabel 19. Contoh Penjadwalan Digambarkan dengan Nama Mapel	70
Tabel 20. Pengalokasian Jam Pelajaran	74
Tabel 21. Parameter Penghitungan Nilai <i>Error</i>	76
Tabel 22. Contoh Hasil Penghitungan Nilai Probabilitas.	77

Tabel 23. Bentuk Struktur Data Probabilitas.	78
Tabel 24. Mekanisme <i>Crossover</i>	82
Tabel 25. Pemrosesan Mutasi	85
Tabel 26. Tabel <i>Class</i> untuk <i>Package</i> Jadwal Algoritma Genetika.....	90
Tabel 27. Jumlah Baris Kode.....	95
Tabel 28. Perbandingan Hasil Pengujian Faktor Kualitas <i>Correctnes</i>	97
Tabel 29. <i>Test case Functionally</i> Modeinput01	98
Tabel 30. Hasil Pengelolaan <i>Test case</i>	98
Tabel 31. Tabel Kriteria Kelolosan.....	99
Tabel 32. Tabel Hasil Uji <i>Functionally</i>	99
Tabel 33. Kategori <i>Usability</i> Skala Likert.	100
Tabel 34. <i>Test Case Portability</i> untuk OS Windows 7.....	101
Tabel 35. Hasil Uji <i>Portability</i>	102

DAFTAR GAMBAR

Gambar 1. Bagan Perulangan Algoritma Genetika.....	26
Gambar 2. <i>Linear Sequential Software Model</i>	30
Gambar 3. Kualitas Perangkat Lunak menurut McCall.....	31
Gambar 4. Prosedur Kerja dan Urutan Kerja.....	40
Gambar 5. Flowchart Algoritma Genetika.....	44
Gambar 6. Penghitungan Reliabilitas Instrumen.	58
Gambar 7. Hasil Penghitungan Reliabilitas	59
Gambar 8. Kromosom.....	65
Gambar 9. Desain Penelitian.....	66
Gambar 10. Hasil <i>output</i> Tabel Jadwal.....	71
Gambar 11. <i>Flowchart Crossover</i>	80
Gambar 12. <i>Flowchart</i> Mutasi.....	84
Gambar 13. Desain GUI untuk Menu <i>Input</i>	86
Gambar 14. Desain GUI untuk Menu Eksekusi.....	86
Gambar 15. Desain GUI untuk Tabel Jadwal	87
Gambar 16. Desain GUI untuk hasil Export XLS	87
Gambar 17. <i>Warning</i> untuk Kesalahan <i>Input</i> Maksimal Generasi	88
Gambar 18. <i>Warning</i> untuk Kesalahan <i>Input Crossover</i>	88
Gambar 19. <i>Warning</i> untuk Kesalahan <i>Input</i> Mutation Rate	88
Gambar 20. <i>Warning</i> untuk Kesalahan <i>Input</i> Bernilai Negatif.....	88
Gambar 21. <i>Warning</i> untuk Kesalahan <i>Input</i> Maksimal Generasi	89

Gambar 22. <i>Warning</i> untuk Belum Mencapai <i>Fitness</i> Maksimal.....	89
Gambar 23. <i>Input</i> Iterasi	91
Gambar 24. Hasil Tabel Jadwal	92
Gambar 25. <i>Warning</i> Keberhasilan Penyimpanan ke xls	92
Gambar 26. Grafik Penurunan Nilai <i>error</i> Terhadap Iterasi pada Percobaan 1. ..	93
Gambar 27. Grafik Penurunan Nilai <i>error</i> Terhadap Iterasi pada Percobaan 2. ..	93
Gambar 28. Grafik Penurunan Nilai <i>error</i> Terhadap Iterasi pada Percobaan 3. ..	94
Gambar 29. Grafik Penurunan Nilai <i>error</i> Terhadap Iterasi pada Percobaan 4. ..	94
Gambar 30. Grafik Penurunan Nilai <i>error</i> Terhadap Iterasi pada Percobaan 5. ..	95
Gambar 31. Hasil Perhitungan <i>Findbugs</i>	96
Gambar 32. Hasil <i>Running Test Portability</i> pada OS Windows 7.....	102

DAFTAR LAMPIRAN

Lampiran Gambar 1. Hasil <i>Portability</i> pada OS Windows 7	115
Lampiran Gambar 2. Hasil <i>Portability</i> pada OS Linux Mint 13 Maya	116
Lampiran Gambar 3. Hasil <i>Portability</i> pada OS Linux Ubuntu	117
Lampiran Gambar 4. Hasil <i>Portability</i> pada OS MACOS Mountain Lion	118
Lampiran Gambar 5. Hasil <i>Portability</i> pada OS MACOS Lion	119
Lampiran Gambar 6. Salinan Validitas Ahli 1.....	152
Lampiran Gambar 7. Salinan Validitas Ahli 2.....	154
Lampiran Gambar 8. Salinan Hasil Angket Pengguna 1.	156
Lampiran Gambar 9. Salinan Hasil Angket Pengguna 2.	158
Lampiran Gambar 10. Salinan Hasil Angket Pengguna 3.	160
Lampiran Gambar 11. Salinan Surat Ijin Penelitian Bappeda.	161
Lampiran Gambar 12. Salinan Permohonan Ijin Penelitian Fakultas.....	162
Lampiran Gambar 13. Salinan Surat Ijin Penelitian Gubernur.....	163
Lampiran Gambar 14. Salinan Surat Keterangan Selesai Penelitian.....	164
Lampiran Tabel 1. <i>Test case</i> ModeInput01	109
Lampiran Tabel 2. <i>Test case</i> ModeInput02	109
Lampiran Tabel 3. <i>Test case</i> ModeRunning01	110
Lampiran Tabel 4. <i>Test case</i> ModeRunning02	110
Lampiran Tabel 5. <i>Test case</i> ModeDokumentasi01	111

Lampiran Tabel 6. <i>Test case</i> ModeDokumentasi02	112
Lampiran Tabel 7. Hasil Pengelolaan <i>Test case</i>	112
Lampiran Tabel 8. Penjabaran Butir Pertanyaan Aspek <i>Usability</i>	113
Lampiran Tabel 9. Hasil Angket <i>Usability</i>	114
Lampiran Tabel 10. <i>Test case</i> ModeWindows.....	115
Lampiran Tabel 11. <i>Test case</i> ModeLinuxMint.....	115
Lampiran Tabel 12. <i>Test case</i> ModelinuxUbuntu.....	116
Lampiran Tabel 13. <i>Test case</i> ModeMac Mountain Lion.....	117
Lampiran Tabel 14. <i>Test case</i> ModeMacOS Lion	118
Lampiran Tabel 15. Kode Program	151

BAB I

PENDAHULUAN

A. Latar Belakang

Sistem penjadwalan mata pelajaran sekolah yang disusun secara konvensional akan dirasa kurang efektif dilakukan ketika terbentur keadaan *input* data banyak dan parameter yang kompleks. Selain membutuhkan ketelitian yang sangat tinggi serta estimasi waktu yang relatif tidak sedikit, metode ini juga memungkinkan terjadinya kesalahan. Sebagai contoh, pertimbangan yang dilakukan untuk menyusun jadwal perlu memperhatikan berbagai komponen yaitu, Guru, Siswa, Ruangan dan Mata Pelajaran.

Dalam rangka menyusun jadwal yang baik, maka harus dilakukan korelasi antar komponen-komponen tersebut agar tidak terjadi kasus “tabrakan” jadwal. Tidak hanya tabrakan jadwal saja yang menjadi pertimbangan. Namun juga beberapa parameter lain, seperti tidak boleh terjadi pengulangan jadwal yang sama dalam satu hari, jumlah jam mengajar guru yang dibatasi, jumlah jam bagi siswa yang disesuaikan dengan tingkatan kelasnya, dan beberapa pelajaran yang tidak boleh dilakukan setelah jam pagi, misalnya pelajaran praktik Olahraga. Dengan banyaknya permasalahan ini, tenaga manusia yang bertugas membuat jadwal tentunya besar kemungkinan akan mendapat kesulitan. Atas dasar kesulitan inilah, peneliti mencoba melakukan pendekatan Algoritma Genetika terhadap sistem penjadwalan mata pelajaran sekolah.

Algoritma Genetika adalah teknik pencarian yang di dalam ilmu komputer untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Algoritma Genetika adalah kelas khusus dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi oleh biologi evolusioner seperti warisan, mutasi, seleksi alam dan rekombinasi (atau *crossover*).

SMA N 1 Kalasan sebagai sekolah menengah atas memiliki problem penjadwalan yang klasik. Di antaranya adanya jadwal yang mengikat, jumlah guru terbatas, dan ada parameter-parameter tertentu. Proses pembuatan jadwal masih dilakukan secara manual dan belum memiliki model optimasi dengan aplikasi komputer.

Melalui teknik Algoritma yang mulai dikembangkan sejak tahun 1970-an di New York Amerika Serikat, satu kesatuan jadwal dipahami sebagai satu individu yang memiliki sejumlah gen. Dimana gen-gen tersebut adalah perwujudan dari susunan kode guru yang diatur dengan *array* tertentu. Pada proses awal, akan dibangkitkan sejumlah individu dengan gen yang random. Setelah itu, akan dibobot kualitas jadwal tersebut sebagai nilai *fitness*. Nilai *fitness* adalah angka yang menyatakan kualitas individu. Semakin individu tersebut berkualitas baik, maka nilai *fitness* pun semakin banyak. Setelah itu, dilakukan proses seleksi menggunakan metode *roulette wheel*. Seleksi ini dimaksudkan untuk mencari individu dengan probabilitas tinggi, dengan *fitness* yang tinggi pula. Pada proses selanjutnya, dilakukan proses *crossover* (kawin silang) dengan cara memotong individu dan memasangkan dengan individu lain. Tidak hanya itu, tahap selanjutnya adalah melakukan mutasi gen. Sebagai gambaran, mutasi gen

dilakukan dengan mengacak nilai random untuk lokasi gen dan mengganti nilai gen di lokasi yang bersangkutan dengan nilai gen lain yang dibangkitkan secara random juga. Dengan selesainya proses-proses di atas, maka didapatkan sejumlah individu baru yang telah mengalami proses seleksi, *crossover*, dan mutasi. Individu-individu ini akan kembali dibobot nilai *fitness* dan jika belum memenuhi syarat *fitness* optimal, maka dilakukan kembali proses sedari awal dengan menggunakan individu-individu baru tersebut. Perulangan akan terus dilakukan hingga mendapatkan individu atau jadwal yang cocok dengan harapan.

B. Identifikasi Masalah

Berdasarkan dari latar belakang di atas, peneliti mengidentifikasi masalah yang terjadi di lapangan sebagai berikut:

1. Pada penjadwalan mata pelajaran sekolah, banyak ditemukan kendala dalam hal ketepatan dan optimasi. Misal terjadinya tabrakan jadwal akibat adanya kelas majemuk dengan jumlah guru dan ruangan yang terbatas.
2. Aspek penyebaran jadwal juga turut berpengaruh terhadap optimasi penjadwalan, di mana tidak dibenarkan terjadi penempatan pelajaran olahraga pada jam tertentu.
3. Tenaga pembuat jadwal terbatas, artinya hanya kelompok petugas atau guru di bagian kurikulum saja yang kerap terlibat dalam pembuatan jadwal.
4. Model penjadwalan yang ada masih berupa model manual yang belum memiliki sistem optimasi.
5. Adanya randomisasi kepentingan setiap tenaga pengajar. Semisal ada guru yang tidak boleh mengajar pada hari tertentu.

6. Adanya mata pelajaran Olahraga yang harus ditempatkan pada slot waktu pagi pada jam ke satu atau kedua.

C. Pembatasan Masalah

Berdasarkan pada masalah di atas, peneliti membatasi permasalahan dalam hal pembuatan sistem optimasi penjadwalan mata pelajaran sekolah menggunakan Algoritma Genetika untuk permasalahan penjadwalan sederhana di SMA N 1 Kalasan dengan optimasi maksimal untuk empat kelas dengan slot guru yang berbeda untuk setiap angkatan. Aplikasi dibuat dengan bahasa Java *desktop* yang digenerate untuk mencari nilai *fitness* yang optimal.

D. Rumusan Masalah

Berdasarkan latar belakang pada poin A, maka dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana cara implementasi Algoritma Genetika terhadap *Software* Sistem Penjadwalan Mata Pelajaran Sekolah?
2. Bagaimana kelayakan pendekatan Algoritma Genetika terhadap masalah penjadwalan mata pelajaran sekolah menurut kaidah *correctness*, *functionally*, *usability* dan *portability*?

E. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui bagaimana cara implementasi pembuatan *Software* Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika.

2. Mengetahui bagaimana kelayakan pendekatan Algoritma Genetika terhadap masalah penjadwalan mata pelajaran sekolah dengan parameter *correctness*, *functionally*, *usability* dan *portability*.

F. Manfaat Penelitian

Penelitian tentang penggunaan Algoritma Genetika untuk Sistem Penjadwalan Mata Pelajaran Sekolah diharapkan dapat bermanfaat baik secara teoritis maupun secara praktis:

1. Secara Teoritis
 - a. Untuk pengembangan ilmu pengetahuan, terutama bagi dunia Teknik Informatika dan Ilmu Komputer.
 - b. Menemukan cara pengimplementasian Algoritma Genetika terhadap *Software* Sistem Penjadwalan Mata Pelajaran Sekolah.
2. Secara Praktis
 - a. Bagi Peneliti
 - 1) Memahami proses kerja Algoritma Genetika dan bentuk implementasinya.
 - 2) Sebagai portopolio untuk peneliti yang berguna untuk masa yang akan datang.
 - b. Bagi Universitas
 - 1) Sebagai bahan referensi untuk penelitian yang akan datang.
 - 2) Sebagai bahan evaluasi bagi universitas dalam mengembangkan keilmuan, dalam hal ini yang berkaitan dengan program berbasis Algoritma Genetika.

BAB II

LANDASAN TEORI

A. Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah (Suyanto, 2005). Sesuai awal konsepnya yaitu genetika, algoritma ini juga mengambil istilah-istilah yang ada dalam ilmu genetika seperti populasi, individu, mutasi, kawin silang dan generasi. Konsep yang ada dalam kaidah genetika ini diterapkan menjadi sebuah algoritma komputasi untuk menyelesaikan masalah yang membutuhkan *constraint* tinggi.

Algoritma Genetika sebagai metode optimasi yang *powerfull* dimungkinkan telah menjadi teknik paling terkenal dalam bidang komputasi evolusioner pada saat ini (Mitsuo Gen, Runwei Cheng, 2000, h. 17).

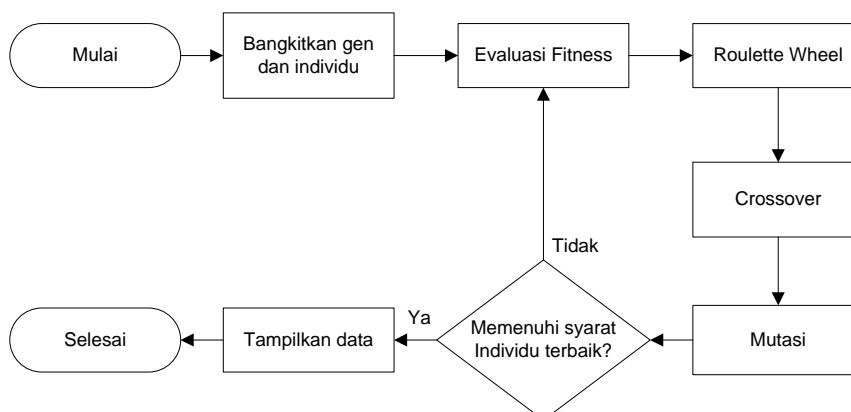
Secara umum, sebuah Algoritma Genetika memiliki lima komponen dasar, seperti yang dilansir dari Michalewicz (1995) ditulis kembali dalam buku “*Genetic Algorithm and Engineering Optimization*” oleh Mitsuo Gen dan Runwei Cheng (Mitsuo Gen, Runwei Cheng, 2000, h. 17): (1) Representasi genetika untuk solusi masalah, (2) Metode menciptakan inisiasi penyelesaian dari populasi, (3) Evaluasi nilai *fitness* berdasarkan kemungkinan solusi, (4) Metode genetika dalam penggantian keturunan dan reproduksi, (5) Hasil akhir yang diharapkan dari pengolahan Algoritma Genetika.

1. Istilah dalam Algoritma Genetika.

Karena mengambil dari kaidah konsep genetika biologi, beberapa istilah dalam Algoritma Genetika juga mengambil konsep yang sama seperti dilansir dari buku berjudul “Algoritma Genetika dalam Matlab” oleh Suyanto (2005), yaitu: 1) Populasi adalah sekelompok individu yang akan dicari penyelesaiannya dalam Algoritma Genetika. 2) Kromosom atau Individu adalah satu permasalahan dan atau penyelesaian yang merupakan komponen iterasi. 3) Gen adalah bagian dari kromosom yang memiliki nilai tertentu dan bertugas menyusun kromosom menjadi individu utuh. 4) Nilai *fitness* adalah derajat kebaikan suatu individu yang menentukan apakah individu tersebut berkualitas atau tidak. 5) Generasi adalah jumlah tingkatan peranakan sebuah kelompok populasi yang akan dikembangkan menjadi populasi yang mutan, mengalami *crossover* dan mutasi beberapa kali sehingga menimbulkan generasi terbaik.

2. Mekanisme Algoritma Genetika

Secara umum, Algoritma Genetika dapat dijabarkan sebagai alur dan bagan pada gambar 1 berikut:



Gambar 1. Bagan Perulangan Algoritma Genetika

Sebelum Algoritma Genetika dijalankan, maka perlu didefinisikan fungsi *fitness* sebagai masalah yang ingin dioptimalkan. Jika nilai *fitness* semakin besar, maka sistem yang dihasilkan semakin baik. Fungsi *fitness* ditentukan dengan metode heuristik. Algoritma Genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode konvensional. Sebagaimana halnya proses evolusi di alam, suatu Algoritma Genetika yang sederhana umumnya terdiri dari tiga operasi yaitu: operasi reproduksi, operasi *crossover* (persilangan), dan operasi mutasi. Struktur umum dari suatu Algoritma Genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

- a. Membangkitkan populasi awal secara random.
- b. Membentuk generasi baru dengan menggunakan tiga operasi di atas (seleksi, *crossover*, mutasi) secara berulang-ulang sehingga diperoleh kromosom yang cukup untuk membentuk generasi baru sebagai representasi dari solusi baru.
- c. Evolusi solusi yang akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom hingga kriteria berhenti terpenuhi.

Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah regenerasi. Beberapa kriteria berhenti yang sering digunakan antara lain:

- a. Berhenti pada generasi tertentu.
- b. Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi/terendah (tergantung persoalan) tidak berubah.

- c. Berhenti bila dalam n generasi berikutnya tidak diperoleh nilai *fitness* yang lebih tinggi/rendah.

3. Seleksi *Roulette*

Seleksi *Roulette* adalah salah satu metode seleksi individu yang tetap melibatkan keanekaragaman populasi. Sesuai dengan namanya, metode ini menirukan permainan *Roulette Wheel* di mana masing-masing individu menempati potongan lingkarannya pada roda *Roulette* secara proporsional sesuai dengan nilai *fitness* (Suyanto, 2005).

4. *Crossover*

Crossover atau kawin silang adalah prosedur memasangkan dua buah individu untuk kemudian dipisahkan masing-masing gennya dan dipasangkan dengan gen pasangannya. Sebuah individu dapat memperoleh solusi yang bagus jika dilakukan proses memindah-silangkan dua buah individu (Suyanto, 2005).

Crossover menyediakan sebuah metode yang memungkinkan terjadinya eksplorasi bagian baru dalam ranah algoritma solusi (Coley, 2000).

5. Mutasi

Dalam dunia nyata, sebuah mutasi dapat terjadi akibat suatu proses. Begitu pula yang terjadi dalam Algoritma Genetika (Coley, 2000). Secara umum, proses mutasi dilakukan dengan cara membangkitkan sebuah bilangan random yang kurang dari probabilitas mutasi (*mutation rate*) kemudian gen yang ada diubah menjadi nilai kebalikannya. Semisal 0 menjadi 1, 1 menjadi 0. Prosedur ini

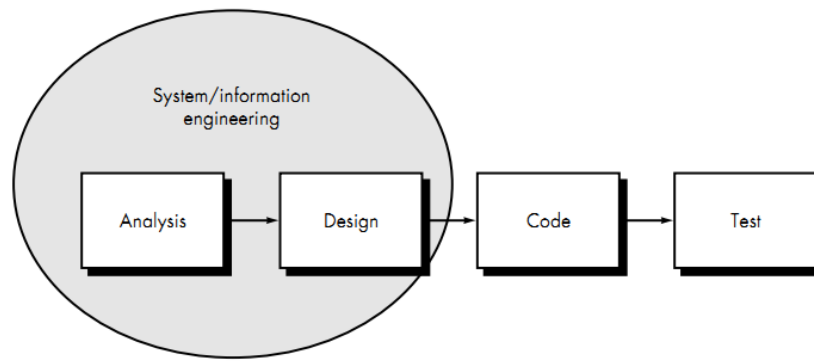
telah disarankan oleh Suyanto dalam bukunya yang berjudul “Algoritma Genetika dalam Matlab” (2005).

B. Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas *platform*, awalnya dikembangkan oleh SUN Microsystem dengan sejumlah keunggulan yang memungkinkan Java dijadikan sebagai bahasa pengembangan *Enterprise* (Fikri, 2005). Akan tetapi, sejak tahun 2010, Java secara resmi dimiliki oleh Oracle Corporation. Bahasa Java memungkinkan penggunaan orientasi objek yang cocok diterapkan untuk Algoritma Genetika. Selain itu, banyaknya dokumentasi dan *source* Algoritma Genetika dalam bahasa Java juga banyak didapat, sehingga memungkinkan untuk pengembangan lebih lanjut.

C. Perancangan Perangkat Lunak

Proses perangkat lunak adalah kerangka kerja yang memuat tugas-tugas yang diperlukan untuk membangun perangkat lunak yang berkualitas baik (Pressman, 2010). Terdapat beberapa teori dalam proses perancangan perangkat lunak, antara lain adalah *Waterfall / Linear Sequential Process*. Proses *Waterfal* merupakan metode *software process* yang paling klasik, dapat disebut juga dengan model sekuensial linier terdiri dari pendekatan sistematis berurutan untuk pengembangan perangkat lunak yang berisi proses analisis, desain, coding, pengujian, dan *support* (Pressman, 2010). Gambar 2 berikut menampilkan *Linear Sequential Model*:

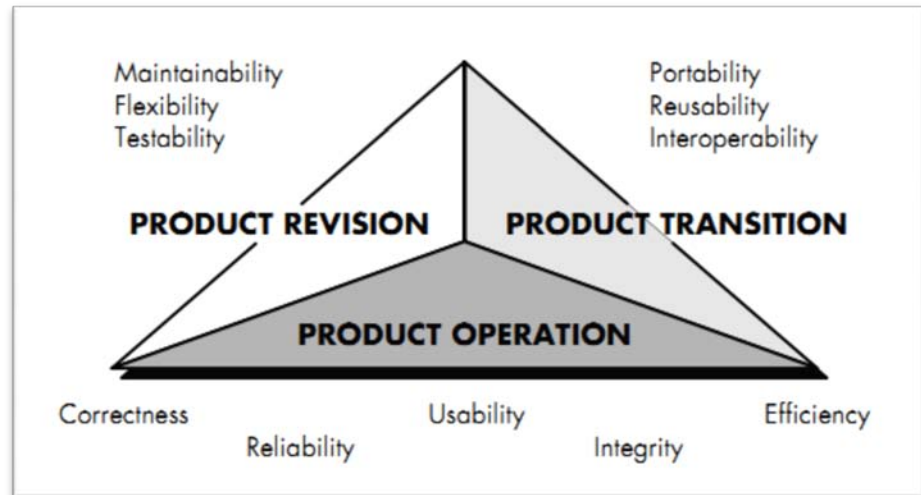


Gambar 2. *Linear Sequential Software Model*.

D. Kualitas Perangkat Lunak (*Software Quality*)

Kualitas Perangkat Lunak atau *Software Quality* adalah gabungan beberapa faktor yang mempengaruhi antara aplikasi dengan konsumen yang membutuhkan. Dalam hal ini, kualitas *software* diidentifikasi sebagai deskripsi dari aktifitas manusia dalam mengapresiasi sebuah *software* (Pressman, 2010).

Terdapat beberapa teori tentang kualitas perangkat lunak, salah satunya adalah *McCall Quality Factor*. McCall, Richards, dan Walter merumuskan serangkaian faktor – faktor yang menunjukkan kualitas perangkat lunak. Faktor – faktor kualitas tersebut dikategorikan menjadi tiga aspek penting dari sebuah perangkat lunak yaitu: karakteristik operasional, kemampuan untuk dalam menangani perubahan, dan kemampuan beradaptasi dengan lingkungan baru (Pressman, 2010). Kategori tersebut digambarkan dalam gambar 3 di bawah ini.



Gambar 3. Kualitas Perangkat Lunak menurut McCall

Faktor – faktor kualitas yang menunjukkan kualitas perangkat lunak tersebut antara lain:

- a. *Correctness* (ketepatan): berkaitan dengan bagaimana program mampu memenuhi spesifikasi dan tujuan yang ingin dicapai oleh pengguna.
- b. *Reliability*: berkaitan dengan bagaimana sebuah program mampu beroperasi dalam sebuah kondisi yang menuntut presisi tertentu.
- c. *Usability* (kebergunaan): berkaitan dengan usaha yang diperlukan pengguna untuk mengoperasikan, menyiapkan *input*, dan menginterpretasikan *output* dari program.
- d. *Integerity* (integritas): berkaitan dengan tingkat kontrol terhadap program oleh pengguna, baik yang mendapatkan otorisasi atau tidak.
- e. *Efficiency* (efisien): berkaitan dengan jumlah sumber daya komputer yang digunakan serta kode yang diperlukan di dalam program untuk menjalankan setiap fungsinya.

- f. *Maintainability*: berkaitan dengan usaha yang diperlukan untuk menemukan dan mengatasi kesalahan di dalam program.
- g. *Flexibility* (Fleksibel): berkaitan dengan usaha yang diperlukan untuk mengubah program yang beroperasi.
- h. *Testability*: berkaitan dengan usaha yang diperlukan untuk menguji sebuah program untuk memastikan bahwa program tersebut berfungsi sebagaimana mestinya.
- i. *Portability*: berkaitan dengan usaha yang diperlukan untuk dapat mentransfer sebuah program dari sebuah lingkungan perangkat keras atau lunak tertentu ke lingkungan yang lain.
- j. *Reusability*: berkaitan dengan bagaimana sebuah bagian program dapat digunakan kembali di dalam program lain.
- k. *Interoperability*: berkaitan dengan usaha yang diperlukan untuk menghubungkan sebuah sistem dengan sistem yang lain.

E. Kualitas *Correctness*

Suatu program harus beroperasi dengan benar dengan parameter sejauh mana perangkat lunak dapat melakukan fungsi yang dibutuhkan. Ukuran yang paling umum untuk menilai faktor *correctness* atau kebenaran adalah *error* per KLOC (*kilo lines of code*), di mana cacat pada program didefinisikan sebagai ketidaksesuaian antara kode dengan persyaratan perangkat lunak. Ketika menilai kualitas keseluruhan produk perangkat lunak, *error* didefinisikan sebagai masalah yang dilaporkan oleh pengguna program setelah program telah dirilis untuk penggunaan umum. (Pressman, 2010).

Terdapat beberapa macam metode pengukur kebenaran per KLOC, di antaranya adalah teori dari McConnell (McConnell, 2004) yang dijabarkan pada tabel 1:

Tabel 1. Perkiraan Jumlah *error* KLOC Teori McConell

Ukuran Project (<i>Line of Code / LOC</i>)	Perkiraan Jumlah <i>error</i>
Lebih kecil dari 2K	0 – 25 <i>error / KLOC</i>
2K – 16K	0 – 40 <i>error / KLOC</i>
16K – 64K	0.5 – 50 <i>error / KLOC</i>
64K – 512K	2 – 70 <i>error / KLOC</i>
Lebih dari 512K	4 – 100 <i>error / KLOC</i>

Selain teori dari McConnell di atas, terdapat pula teori metode pengukuran kebenaran kode per KLOC versi Pressman (Pressman, 2010, h. 709) yang dijabarkan pada tabel 2:

Tabel 2. Perkiraan Jumlah *Error* KLOC Teori Pressman

No	Metode	Rumus
1	Watson – Felix Model	$E = 5.2 \times (KLOC)^{0.91}$
2	Bailey-Basili Model	$E = 5.5 + 0.73 \times (KLOC)^{1.16}$
3	Boehm Simple Model	$E = 3.2 \times (KLOC)^{1.05}$
4	Doty Model(untuk $KLOC > 9$)	$E = 5.28 \times (KLOC)^{1.047}$

F. Kualitas *Functionally*

Functionally merupakan faktor kualitas yang menunjukkan tingkat kemampuan menyediakan fungsi – fungsi yang diharapkan sehingga dapat memberikan kepuasan kepada pengguna (Pressman, 2010).

Faktor kualitas *functionally* dapat diuji dengan analisis fungsionalitas dari setiap komponen dari suatu perangkat lunak. Terdapat beberapa metode pengujian fungsionalitas. Salah satunya adalah metode *black-box testing* yang merupakan merupakan metode yang cocok untuk melakukan pengujian fungsionalitas perangkat lunak. Dalam bukunya, Pressman (2010) menjelaskan bahwa *black-box testing*, atau juga disebut *behavioral testing*, adalah *testing* yang terfokus pada kebutuhan fungsional dari suatu perangkat lunak. Pengujian ini memungkinkan analis sistem memperoleh kumpulan kondisi *input* yang akan mengerjakan seluruh keperluan fungsional program.

James Bach dalam tulisanya “*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo Desktop Applications Edition*” (2005, h. 4) membagi fungsi dalam sebuah perangkat lunak menjadi dua yaitu: *primaryfunction* (fungsi primer) dan *contributingfunction* (fungsi pendukung). Fungsi primer merupakan fungsi yang utama dalam perangkat lunak, kesalahan dalam fungsi ini akan membuat perangkat lunak tidak layak atau tidak dapat digunakan. Sedangkan fungsi pendukung merupakan fungsi yang memberikan kontribusi pada perangkat lunak, tetapi bukan merupakan fungsi utama.

Dalam kaitanya dengan standar yang digunakan untuk menentukan apakah sebuah perangkat lunak lolos dalam pengujian faktor kualitas *functionality*, (Bach, 2005, h. 5) dalam tulisanya yang berjudul “*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo*” memberikan gambaran bagaimana suatu perangkat lunak dapat dikatakan memenuhi faktor kualitas *functionally* dalam program *Windows Logo Certification*. Tabel 3 di bawah ini menunjukkan kriteria kelolosan:

Tabel 3. Kriteria Kelolosan dalam Aspek *Functionally*.

Kriteria Lolos	Kriteria Gagal
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya.
2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

G. Kualitas *Usability*

Usability adalah suatu usaha untuk mengukur tingkat kebergunaan sebuah program dan dapat diukur dari empat karakteristik: (1) keterampilan dan tingkat intelektual yang diperlukan untuk belajar sistem, (2) waktu yang dibutuhkan untuk penggunaan sistem secara efisien, (3) peningkatan produktivitas diukur ketika sistem ini

digunakan oleh seseorang yang cukup efisien, dan (4) penilaian subyektif (dapat juga diperoleh melalui kuesioner) atau sikap pengguna terhadap sistem (Pressman, 2010).

Standar ISO 9126 mengkategorikan *usability* sebagai faktor kualitas nonfungsional. *Usability* berkaitan langsung dengan bagaimana sebuah perangkat lunak digunakan oleh pengguna. Standar ISO 9126 membagi faktor kualitas *usability* menjadi beberapa subfaktor yaitu *understandability*, *learnability*, *operability* dan *attractiveness* (Hass, 2008).

Understandability berkaitan dengan tingkat kesulitan pengguna dalam mengerti bagaimana menggunakan perangkat lunak dalam konsep logis. *Learnability* berkaitan dengan bagaimana pengguna dapat belajar dalam menggunakan suatu perangkat lunak. *Operability* berkaitan dengan bagaimana pengguna dapat menggunakan fungsi – fungsi dalam perangkat lunak. Sementara *attractiveness* berhubungan dengan bagaimana kemenarikan perangkat lunak sehingga pengguna mau menggunakannya (Hass, 2008).

Anne Mette Jonassen Hass menjelaskan (2008) menjelaskan bahwa faktor kualitas *usability* dapat diuji dengan metode survey atau kuesioner. Metode survey atau kuisisioner digunakan untuk menganalisis faktor kualitas *usability* dari sisi subyektif pengguna. Pertanyaan – pertanyaan yang digunakan dalam kuisisioner harus mencerminkan persepsi pengguna terhadap perangkat lunak yang dikembangkan. Pertanyaan – pertanyaan tersebut juga seharusnya mencakup pada sub faktor kualitas *usability* yaitu *understandability*, *learnability*, *operability* dan *attractiveness* (Hass, 2008, h. 254).

Jacob Nielson (Nielsen, 1993) mengemukakan bahwa faktor kualitas *usability* atau kebergunaan adalah atribut kualitas yang menunjukkan seberapa mudah suatu antarmuka digunakan. Atribut ini terdiri dari lima sub komponen yaitu *learnability*, *efficiency*, *memorability*, *errors*, *kepuasan*.

H. Kualitas *Portability*

Faktor kualitas *portability* menggambarkan kemampuan perangkat lunak untuk dapat dipindah dan dijalankan di lingkungan yang berbeda, dalam kaitanya dengan penelitian ini adalah pada komputer dengan spesifikasi *hardware* maupun *operating system* yang berbeda – beda (Pressman, 2010).

Bahasa pemrograman Java merupakan bahasa dengan keunggulan pada aspek *portability* dan *architecture neutral*. Aplikasi yang dibuat dengan bahasa pemrograman java dapat berjalan di berbagai *platform* berbeda, karena Java bersifat *multiplatform* (Fikri, 2005).

Untuk menjalankan aplikasi Java pada komputer diperlukan Java Runtime Environment (JRE). Java Runtime Environment tersedia secara gratis berbagai sistem operasi: Windows, Linux, Macintosh, dan Solaris. Dalam kaitanya dengan *hardware requirement* (kebutuhan perangkat lunak), Java Runtime Environment tidak membutuhkan *hardware* dengan spesifikasi yang tinggi. Berdasarkan dokumen dalam *website* resminya, JRE hanya membutuhkan Processor dengan kecepatan 233 Mhz dan RAM sebesar 126 MB (pada Sistem Operasi Windows dan Macintosh) / 64 MB (Pada Sistem Operasi Linux dan Solaris). Secara umum dapat dikatakan bahwa semua komputer dengan sistem

operasi yang didukung Java Runtime Environment dapat menjalankan aplikasi Java.

Pengujian faktor kualitas *portability* dapat dilakukan dengan mencoba aplikasi Java pada *enviromtment* yang berbeda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda.

I. Penelitian yang Relevan

1. Gibbon Mardame Parsaroan Tamba (2004), Sistem Penjadwalan Matakuliah menggunakan Algoritma Genetika (Studi Kasus Fakultas Matematika dan IPA IPB). Dalam penelitian ini, dilakukan metode perbandingan penyusunan jadwal secara manual dengan penyusunan jadwal menggunakan Algoritma Genetika pada jadwal kuliah Fakultas MIPA IPB. Kesimpulan dari penelitian ini adalah metode Algoritma Genetika dinyatakan efektif untuk menyusun jadwal perkuliahan secara optimal.
2. Sam'ani (2012), Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika. Penelitian ini mengambil kasus penjadwalan perkuliahan di STMIK Palangkaraya. Dimana hasil yang didapat adalah penjadwalan dinilai menjadi lebih optimal setelah iterasi yang relatif pendek, yaitu 5-10 generasi.
3. Algoritma Genetika telah digunakan untuk menghasilkan nilai maksimum luas *coverage area* dan biaya minimum operasional penempatan armada kapal TNI AL di kawasan timur Indonesia. Sistem yang dibuat diuji dengan penempatan armada kapal sebanyak 27 buah untuk ditempatkan pada 28 pangkalan yang ada. Hasil yang didapatkan adalah luas *coverage area* 1.942.929 mil² dan

biaya operasional Rp 2.853.447.000 dengan nilai *fitness* terbaik 6,6330. Jika dibandingkan dengan data lapangan yang ada yaitu total luas area yang harus diamankan sekitar 1.688.765 mil² dengan budget biaya Rp 5.000.000.000 maka hasil yang didapatkan lebih efektif dan efisien (Sam'ani, 2012).

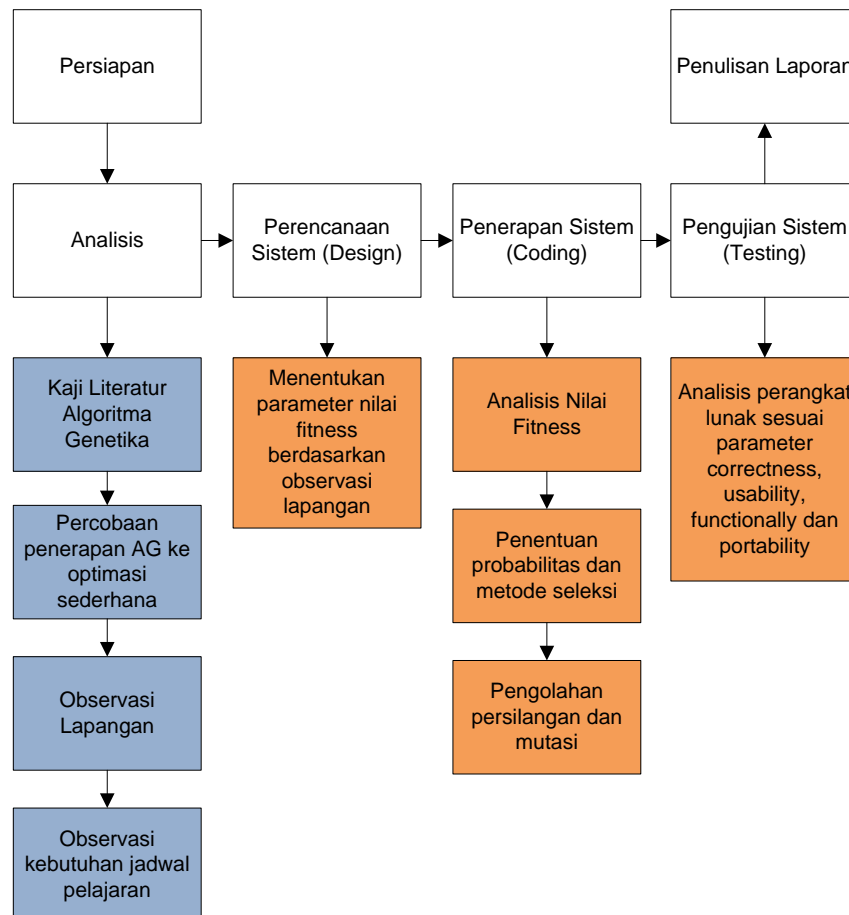
4. Algoritma Genetika juga telah digunakan untuk pengolahan pengenalan citra huruf kapital dengan metode pembangkitan nilai *fitness* berdasar transformasi digital seperti translasi, dilatasi dan rotasi. Dalam penelitian tersebut dihasilkan pengenalan yang optimal dengan generasi di atas 2000 (Saputro, 2003).
5. Sebagai sistem pendukung keputusan teknik perdagangan dalam masalah TSP (*Traveling Salesman Problem*). Metode ini telah diteliti oleh Franklin Allen dan Risto Karjalainen dalam jurnal yang berjudul “*Using Genetic Algorithm to Find Technical Trading*” (Allen, 1999).

BAB III

METODOLOGI PENELITIAN

A. Model dan Metode Penelitian

Mengembangkan dari model tahap pembuatan perangkat lunak seperti pada metode linear, prosedur dan urutan kerja dalam penelitian ini dapat dijabarkan menjadi beberapa sub prosedur pada gambar 4 sebagai berikut:



Gambar 4. Prosedur Kerja dan Urutan Kerja

Tahap pertama adalah persiapan yang meliputi perencanaan awal dalam bentuk rencana penelitian. Rencana ini berisi langkah-langkah penelitian secara umum. Yang kedua adalah Observasi Lapangan & Kaji Literatur Algoritma Genetika (Analisis) yang dibutuhkan untuk memperoleh gambaran dari objek penelitian yaitu jadwal pelajaran. Untuk keperluan ini, observasi dilakukan di salah satu SMA di Kabupaten Sleman yaitu SMA N 1 Kalasan. Kemudian kaji literatur dilakukan dengan memperkaya bacaan tentang Algoritma Genetika, baik melalui media internet, jurnal, penelitian kontemporer, maupun buku-buku yang relevan. Kemudian tahap selanjutnya adalah Perancangan Sistem (*Design*), Penerapan Sistem (*Coding*), Pengujian Sistem (*Testing*), dan Penulisan Laporan.

B. Sumber Informan

Sumber informan dalam uji kualitas *usability* adalah personal pembuat jadwal di sekolah terkait (SMA N 1 Kalasan) dan guru TIK yang memahami alur aplikasi. Pemilihan sumber informan ini dititikberatkan pada orang yang memang berkecimpung dalam dunia penjadwalan sekolah (personal kurikulum atau pembuat jadwal) dan guru TIK yang memiliki pemahaman terhadap aplikasi.

C. Lokasi Penelitian

Lokasi penelitian untuk uji kemampuan dan pendekatan Algoritma Genetika untuk penjadwalan mata pelajaran sekolah dilakukan secara individu di PC peneliti. Dalam hal ini pengujian terhadap kualitas dan kenaikan nilai *fitness*, kesesuaian fungsi objektif, dan kualitas *software correctness*, *functionally* dan *portability*.

Untuk melakukan uji *software* jadwal sekolah untuk faktor *usability*, peneliti mengambil observasi di salah satu sekolah menengah atas di Sleman, yaitu SMAN 1 Kalasan, Sleman. Berbagai kendala dan keadaan lapangan dikaji untuk menentukan bagaimana desain algoritma dan fungsi objektif Algoritma Genetika dijalankan.

D. Teknik Pengumpulan Data

1. Teknik Pengaplikasian Algoritma Genetika.

Secara singkat, istilah-istilah dan bagian Algoritma Genetika dalam aplikasinya sebagai perangkat lunak penjadwalan dapat dijelaskan pada tabel 4 berikut:

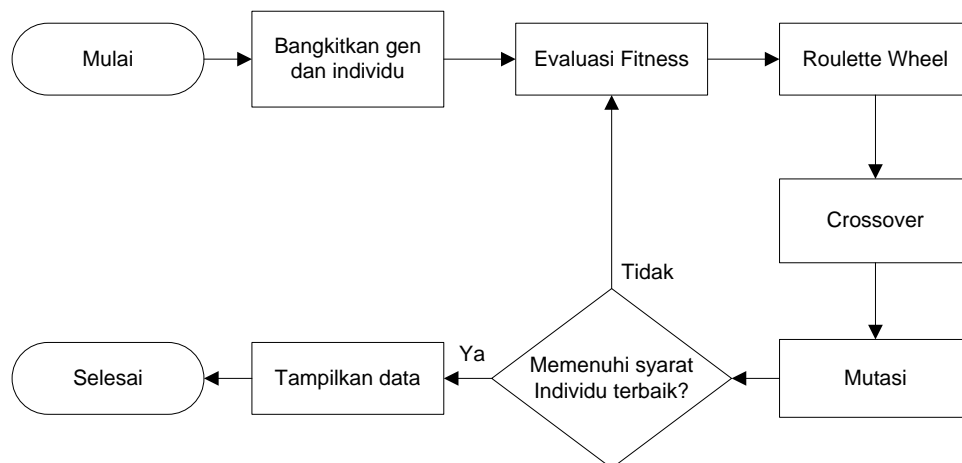
Tabel 4. Istilah dalam Algoritma Genetika

Nama Istilah	Penjelasan
Populasi	Populasi adalah jumlah jadwal acak yang dibangkitkan dengan jumlah tertentu. Misal, jadwal 1 dan jadwal 2 dikatakan sebagai satu kesatuan populasi.
Kromosom = Jadwal	Satu kesatuan jadwal dinamakan individu yang memiliki identitas kromosom yang berbeda-beda. Panjang kromosom dipengaruhi oleh panjang gen, yaitu banyaknya pertemuan dalam satu jadwal.
Gen = pertemuan	Tiap pertemuan disebut gen untuk kromosom jadwal.
Iterasi	Iterasi adalah bilangan yang menunjukkan tingkat generasi dan perulangan.
<i>Crossover</i>	<i>Crossover</i> adalah proses kawin silang di mana sepasang individu dipotong melalui matriks tertentu untuk dipasangkan dengan individu pasangannya.
Mutasi	Mutasi adalah penggantian satu struktur gen di dalam individu yang dibangkitkan secara random.

a. Gambaran Umum Metode

Metode pada Algoritma Genetika memiliki beberapa tahap dan sekian perulangan. Dalam kasus ini, metode pada pembuatan jadwal kelas memiliki beberapa tahap seperti pada gambar 5 yaitu:

- 1) Pembangkitan individu.
- 2) Evaluasi fungsi objektif dan nilai *fitness*.
- 3) Penentuan nilai probabilitas.
- 4) Seleksi *Roulette Wheel*.
- 5) *Crossover* / Persilangan.
- 6) Mutasi
- 7) Pembangkitan individu baru.



Gambar 5. Flowchart Algoritma Genetika

b. Variabel yang Digunakan dalam Proses Algoritma Genetika

1) Jumlah Individu per Populasi

Jumlah individu per populasi ditentukan sendiri pada awal pemrograman. Semakin banyak jumlah individu per populasi, semakin besar kemungkinan individu yang unggul dan mampu berregenerasi dengan cepat dan efisien.

2) Jumlah Generasi (iterasi)

Jumlah generasi adalah jumlah maksimal iterasi / paket perulangan yang diperbolehkan. Variabel ini menentukan sampai berapa kali populasi awal akan berubah, jadi juga memiliki peran yang tak kalah penting dalam menampilkan jumlah variasi individu, yang akan berpengaruh terhadap hasil Algoritma Genetika. Dalam kasus ini dibatasi maksimal 100 generasi untuk mendukung memori yang terdapat dalam komputer yang digunakan dalam melakukan Algoritma Genetika ini. Jadi hal itu berhubungan erat dengan keterbatasan memori yang terdapat dalam sebuah komputer.

3) Fungsi Objektif dan Nilai *Fitness*

Fungsi objektif adalah formula yang dibuat untuk mengukur derajat kualitas individu. Formula ini yang akan dibuat untuk menghitung angka nilai *fitness*. Nilai *fitness* adalah bilangan yang menunjukkan kualitas individu. Semakin tinggi nilai *fitness*, semakin tinggi kualitas dan semakin tinggi pula tingkat probabilitas seleksi.

4) Tingkat Persilangan (*cross over*)

Tingkat persilangan adalah peluang untuk terjadi persilangan antara sepasang individu. Dalam kenyataannya persilangan akan selalu terjadi, hanya saja jumlah gen dan gen-gen yang disilangkan akan berbeda-beda. Oleh karena itu, biasanya TP diisi dengan nilai 1 atau 100%, yang berarti akan selalu terjadi persilangan.

5) Persentase Persilangan (*crossover rate*)

PP merupakan angka persentase yang akan mempengaruhi berapa gen yang akan disilangkan. Misalkan dalam satu individu, ada 6 gen dan $PP=0.5$. Maka jika terjadi persilangan, gen-gen yang akan disilangkan sejumlah 3 buah gen.

6) Tingkat Mutasi (*mutation rate*)

Variabel yang berupa angka persentase ini akan mempengaruhi cukup banyak terjadinya mutasi dalam suatu populasi. Variabel TM merupakan salah satu variabel yang berbentuk peluang, artinya kemungkinan terjadinya mutasi dilihat tiap individunya. Misalkan JI 10, angka TM diisi 0,5, bukan berarti akan ada 5 mutasi (0.5×10), tetapi peluang terjadinya mutasi untuk masing-masing individu adalah 0.5. Jadi dalam contoh diatas bisa saja terjadi 0 mutasi sampai 10 mutasi.

c. Pembangkitan Populasi Individu

Populasi merupakan kumpulan beberapa individu. Semua populasi dalam Algoritma Genetika ini berasal dari satu populasi yaitu populasi awal. Solusi atau kromosom terbaik dari populasi awal ini akan dipertahankan, dan akan

mengalami proses evolusi untuk mendapatkan kemungkinan solusi yang lebih baik.

Pembuatan populasi awal ini dilakukan melalui proses pemilihan secara acak dari seluruh solusi yang ada. Pemilihan acak ini menyebabkan populasi awal dari Algoritma Genetika tidak akan sama dalam setiap kali percobaan, meskipun semua nilai variabel yang digunakan sama.

Dalam hal ini populasi adalah sebuah *string* yang berisi gen yang berjumlah sesuai dengan jumlah praktikum. Sebagai contoh apabila terdapat suatu gen yang panjangnya 3 karakter, maka panjang *string* adalah $3 \times n$, dengan n merupakan jumlah mata pelajaran yang akan di-*generate*.

Semua kelas yang akan disimpan pada palet-palet, terlebih dahulu akan disimpan pada sebuah *array*. *Array* tersebut berisi informasi urutan kelas tersebut pada *array*, dan posisinya ketika akan dimasukkan ke dalam palet. Adapun skemanya dapat dilihat pada bagan berikut ini:

Pertemuan[i] = [kodemapel[i], kodemapel[i],];

Pertemuan[i] = [kodemapel[i], kodemapel[i],];

Dst.

Misalnya dalam satu minggu diwajibkan ada pertemuan pelajaran Matematika sebanyak 2 jam pertemuan untuk masing-masing kelas. Dan dalam satu minggu terdapat 25 pilihan waktu dari Senin hingga Sabtu.

Hal ini berlaku untuk masing-masing mata pelajaran. Dan nilai permutasi ditentukan sebanyak jumlah pertemuan yang diinginkan.

d. Evaluasi Nilai *Fitness*

Untuk mengetahui baik tidaknya solusi yang ada pada suatu individu, setiap individu pada populasi harus memiliki nilai pembandingnya (*fitness cost*). Melalui nilai pembanding inilah akan didapatkan solusi terbaik dengan cara pengurutan nilai pembanding dari individu-individu dalam populasi. Solusi terbaik ini akan dipertahankan, sementara solusi lain diubah-ubah untuk mendapatkan solusi yang lain lagi, melalui tahap *cross over* dan mutasi (*mutation*).

Sebelum melakukan penempatan jadwal kelas dilakukan dua buah pengecekan terlebih dahulu, yaitu pencarian hari dan jam yang masih kosong dan pengecekan prioritas yaitu pada hari dan jam mana yang paling tinggi prioritasnya. Proses evaluasi ini melibatkan fungsi objektif yang merupakan formula untuk menentukan jumlah nilai *error* dan dikalkulasikan dengan nilai *fitness*.

Adapun evaluasi nilai *fitness* dilakukan dengan parameter sebagai berikut:

1. Tidak boleh terjadi tabrakan jadwal.
2. Tidak boleh terjadi penumpukan ruangan.
3. Mata pelajaran olahraga tidak boleh di atas jam ke 2.
4. Tidak boleh terjadi penumpukan jadwal yang sama dalam satu hari.

Apabila terdapat aturan-aturan yang dilanggar maka nilai *fitness cost* akan dikurangi sehingga hasilnya akan menjadi lebih jelek. Contoh aturan tentang *fitness cost* akan dijelaskan pada tabel 5 berikut:

Tabel 5. Contoh Tabel *Fitness Cost*

Aturan	<i>Fitness cost</i>
Jadwal bertabrakan	<i>Fitness cost</i> – (jumlah mata kuliah bentrok *100)
Ruangan bertabrakan	<i>Fitness cost</i> – (jumlah responsi bentrok *100)
Mapel olahraga di atas jam ke 2	<i>Fitness cost</i> – (jumlah matakuliah semester *10)
Penumpukan jadwal yang sama dalam sehari	<i>Fitness cost</i> – (jumlah praktikum tidak sesuai *100)

e. Penentuan Nilai Probabilitas

Penentuan nilai probabilitas didasarkan pada nilai *fitness* yang ditentukan pada poin sebelumnya. Nilai *fitness* akan ditotal dari semua iterasi pembangkitan dan dikalikan angka 1000 sebagai random probabilitas. Contoh penentuan nilai probabilitas dapat dilihat pada *source* pada tabel 6 di bawah ini:

Tabel 6. Sintaks Java untuk Penentuan Probabilitas.

```

public double evaluateawal() {
    double total = 0;
    double finest = 0;
    for (int i = 0; i<SIZE; i++) {
        total += this.nilaifungsi[i];
        finest += this.fitness[i];
    }
    setTotalfitness(finest);
    System.out.println("Total nilai fungsi = "+total);
    System.out.println("Total nilai fitness = "+finest);
    for (int i = 0; i<SIZE; i++) {
        double probability;
        probability =
        (this.getFitness(i)/this.getTotalfitness()*1000;
        this.setProbabilitas(i, probability); //set
        probabilitas untuk proses seleksi
    }
    return total;
}

```

f. Seleksi *Roulette Wheel*

Melanjuti tahap sebelumnya yaitu penentuan probabilitas, maka yang dilakukan adalah seleksi *fitness*. Dalam kasus ini yang digunakan adalah seleksi menggunakan *Roulette Wheel* (Mesin *Roulette*) dimana setiap individu yang

memiliki nilai *fitness* tinggi akan memiliki nilai probabilitas untuk terpilih yang tinggi. Adapun bentuk bagannya adalah sebagai berikut:

- 1) Dihitung nilai *fitness* masing-masing individu.
- 2) Dihitung total *fitness* semua individu.
- 3) Dihitung probabilitas tersebut untuk jatah 1 – 1000.
- 4) Dibangkitkan bilangan acak 1 – 1000.
- 5) Dari bilangan acak yang dihasilkan, ditentukan mana individu yang terpilih dalam proses seleksi.

Contoh *Roulette Wheel* dapat dilihat pada *source* tabel 7 di bawah ini:

Tabel 7. Contoh Sintaks Java untuk *Roulette Wheel*.

```

public void RouletteWheel() {
    for (int i = 0; i < GA_1Individu.SIZE ; i++) {
        int indexProb = (int) (Math.random()*1000);
        System.out.println("Hasil index seleksi
Roulette      "+i+"      =      "+indexProb+"      index      individu      =
"+indexuntukRolet[indexProb]);
        setHasilRolet(i, indexuntukRolet[indexProb]);
    }
}

```

g. *Crossover* (Persilangan)

Setelah menjalani proses seleksi, maka individu yang terpilih akan dilakukan *Crossover*. Tahapan ini akan menyilangkan dua individu yang ada dalam suatu populasi, untuk mendapatkan dua individu baru. Setelah tahap, maka akan didapat populasi baru yang jumlahnya 2 kali lipat dari populasi lama.

Pada kasus penyusunan jadwal pelajaran ini, yang menjadi individu adalah satu urutan penyusunan jadwal pelajaran dalam satu minggu. Adapun contohnya ada pada tabel 8.

Tabel 8. Contoh Individu Penjadwalan.

Pertemuan	Kelas							
	X1	X2	X3	X4	XI IPA 1	XI IPA 2	...	Dst
Senin1	Mat1		Fis1					
Senin2	Fis1	Mat1						
Senin3	KN1	Fis1	Mat1					
Senin4	Bio1							
Selasa1	Seni1							
Selasa2								
Selasa3								
...
Sabtu4								

Individu 1 = 001 002 003 004

Individu 2 = 005 006 007 008

Keterangan:

Mat1 = 001, Fis1 = 002, KN1 = 003, dst

Dari populasi yang ada, diambil individu sepasang demi sepasang untuk disilangkan (*cross over*). Persilangan pada kasus ini dilakukan dengan memindahkan sebagian urutan pada satu individu dan menukarkannya dengan individu yang lain. Ada 2 macam cara yaitu dengan *Two Points* dan *Uniform*. Pada *Two Points Cross Over*, dipilih secara acak 2 titik yang akan disilangkan.

Individu 1 = 001 | 002 003 | 004

Individu 2 = 005 | 006 007 | 008

Sedangkan pada *Uniform Cross Over* ada penentuan persentase gen yang akan disilangkan misalkan 50%, angka ini nantinya perlu masukkan dari *user*.

Individu 1 = 001 002 | 003 004

Individu 2 = 005 006 | 007 008

Setelah disilangkan, akan dilakukan pengecekan terhadap masing-masing individu, apakah terjadi pengulangan. Kedua individu yang telah disilangkan ini diperbaiki, sehingga tidak ada pengulangan lagi. Garis besarnya adalah setiap angka yang diulang ditukar dengan pasangannya, yaitu angka yang diulang di kromosom pasangannya.

Hasil untuk *Two Points Cross Over*:

Individu 1 = 001 006 007 004

Individu 2 = 005 002 003 008

Hasil untuk *Uniform Cross Over*:

Individu 1 = 001 002 007 008

Individu 2 = 005 006 003 004

Populasi baru yang dihasilkan ini akan dibandingkan dengan populasi terbaik yang telah ada dan bila populasi terbaru lebih baik nilai *fitness cost*-nya maka populasi ini akan menggantikan populasi terbaik yang pernah ada.

h. Mutasi

Cara lain untuk mendapatkan individu yang baru yaitu dengan mutasi. Probabilitas terjadinya mutasi gen pada suatu kromosom sangatlah kecil, karena itu dalam penerapannya pada Algoritma Genetika, probabilitasnya seringkali dibuat kecil, lebih kecil dari $\frac{1}{2}$ (*mutation rate*). Berbeda dengan tahap *cross over*, dimana satu individu perlu individu yang lain, pada tahap ini tidak membutuhkan individu yang lain untuk bermutasi. Dalam kasus ini dimungkinkan terjadi 2 macam mutasi dimana probabilitas terjadinya mutasi akan ditentukan *user*.

Mutasi pertama yang mungkin terjadi adalah perubahan urutan kelas praktikum. Hal ini dilakukan secara acak, diambil 2 angka (nomor kelas praktikum) dari satu individu, kemudian ditukar.

Contoh:

Sebelum mutasi : 001 002 003 004

Hasil : 003 002 001 004

i. Pembangkitan Populasi Baru

Satu populasi baru telah terbentuk dengan selesainya mutasi. Populasi baru tersebut akan menjadi populasi awal bagi generasi selanjutnya dan Algoritma Genetika akan mengulang tahap 2 sampai 4 secara terus menerus sampai sejumlah generasi yang telah ditentukan.

2. Instrumen Penelitian

Instrumen penelitian yang digunakan adalah metode kuisioner yang tervalidasi oleh ahli. Kuisioner ini disusun berdasarkan variabel yang ditentukan dalam pengujian. Tahap-tahap penyusunan instrumen penelitian adalah pemahaman yang mendalam tentang variabel atau hubungan antar variabel dan menjabarkannya dalam bentuk sub variabel, indikator, deskriptor dan butir-butir pertanyaan dalam angket, dalam daftar cocok atau pedoman wawancara serta pedoman observasi (Arikunto, 2010).

a. Instrumen untuk Varibel *Correctness*

Untuk melakukan uji *correctness*, peneliti menggunakan instrumen Findbugs. FindBugs merupakan *freeware tools* yang dikembangkan oleh The University of Maryland. Tools ini digunakan untuk menganalisis *error* yang ditemukan dalam *source code* aplikasi yang dikembangkan. FindBugs mengkategorikan jenis *bugs* menjadi beberapa kategori yaitu: *bad practice*, *correctness*, *multithreaded correctness*, *experimental*, *internationalization*, *malicious code vulnerability*, *performance*, *security*, *dodgy code*. Dalam penelitian ini akan digunakan dua kategori yaitu kategori bugs yang berkaitan dengan analisis faktor kualitas *correctness*: *correctness* dan *multithreadedcorrectness*.

b. Instrumen untuk Varibel *Functionally*

Untuk melakukan uji *black-box testing* untuk kualitas *functionally*, peneliti menggunakan instrumen *test case*. Menurut Agarwal, Tayal dan Gupta (2010) *test case* merupakan seperangkat instruksi yang didesain untuk mengetahui kesalahan yang dalam perangkat lunak.

Tujuan dari penggunaan *test case* adalah memberi mekanisme yang dapat membantu memastikan kelengkapan pengujian dan memberikan kemungkinan tertinggi untuk mengungkap kesalahan pada perangkat lunak (Ladjamudin, 2006, h. 358).

Menurut Ladjamudin (2006), semua produk yang direkayasa dapat diuji dengan satu atau dua cara:

- 1) Dengan mengetahui fungsi yang ditentukan di mana produk dirancang untuk melakukannya, pengujian dapat dilakukan untuk memperlihatkan bahwa masing-masing fungsi beroperasi sepenuhnya, pada waktu yang sama mencari kesalahan pada setiap fungsi (uji *black box*).
- 2) Dengan menguji kerja internal suatu produk, maka pengujian dapat dilakukan untuk memastikan bahwa “semua roda gigi berhubungan” yaitu operasi internal bekerja sesuai dengan spesifikasi dan semua komponen internal telah diamati dengan baik (uji *white box*).

Dalam penelitian ini, peneliti menggunakan metode *black box* dengan metode *test case* yang berisi beberapa alur fungsional sebuah sistem. Secara standar, alur *test case* dibuat untuk memperoleh data fungsionalitas program tersusun dari tujuan tes, langkah pengetesan, hasil yang diharapkan, dan hasil final uji alur.

Format *test case* dibuat seragam untuk masing-masing alur. Hal ini difungsikan agar dokumentasi *test case* menjadi lebih terarah. Adapun format standar untuk *test case* seperti yang diusulkan oleh Agarwal, Tayal dan Gupta (2010) adalah seperti tertampil pada tabel 9 berikut:

Tabel 9. Tabel *Test case* untuk Uji *Functionally*.

Test case id	Nama yang unik untuk identifikasi <i>test case</i>
Purpose	Tujuan dari <i>test case</i>
Assumptions	Syarat kondisi awal yang harus terpenuhi sebelum <i>test</i> dapat dijalankan.
Test data	Variabel atau kondisi yang akan di <i>test</i> .
Steps	Langkah – langkah yang dijalankan.
Expected result:	Hasil yang seharusnya didapatkan (yang menunjukkan bahwa tidak ada kesalahan dalam perangkat lunak)
Actual result:	Hasil yang didapat dalam pengujian.
Pass/Fail:	Keterangan: Lolos atau Gagal.

c. Instrumen untuk Varibel *Usability*

Instrumen untuk *usability* dinilai berdasarkan parameter menurut ISO 9126 yang dipaparkan dalam buku “*Software Engineering*” oleh Pressmann (2010) yang menyatakan kualitas *usability* dilihat dari indikator-indikator sebagai berikut:

- 1) *Learnability*: Sistem harus mudah untuk dipelajari oleh user.
- 2) *Understandability*: Sistem harus mudah dipahami.
- 3) *Operability*: Sistem harus mudah dioperasikan.
- 4) *Attractiveness*: Sistem harus menarik untuk pengguna.

Keempat butir sub indikator *usability* tersebut dijabarkan menjadi beberapa butir pertanyaan. Menurut Arikunto (2010) , penahapan yang harus dilalui oleh penyusun instrumen penelitian secara umum adalah:

- 1) Mengidentifikasi variabel penelitian.
- 2) Menjabarkan variabel menjadi sub variabel.
- 3) Menjabarkan sub variabel menjadi deskriptor.

- 4) Memecah deskriptor menjadi indikator.
- 5) Mengubah atau merumuskan indikator ke dalam butir-butir pertanyaan atau butir soal.

Adapun hasil dari penjabaran keempat sub indikator *usability* tersebut dijabarkan pada tabel 10 sebagai berikut:

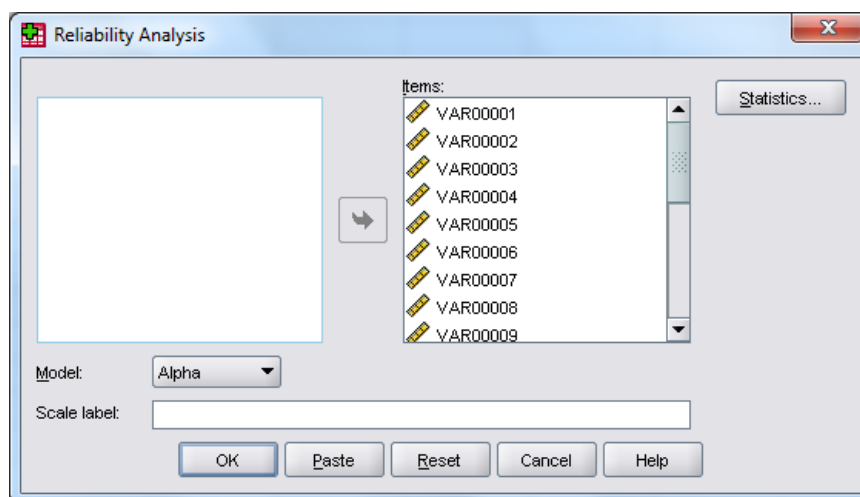
Tabel 10. Penjabaran Butir Pertanyaan Aspek *Usability*.

No	Butir Pertanyaan	Sub Indikator			
		U	L	O	A
1	Apakah kegunaan program dapat dikenali dari awal interface?				
2	Apakah bahasa yang ditampilkan mudah dipahami?				
3	Apakah hasil <i>output</i> tabel penjadwalan mudah dipahami?				
4	Apakah pesan <i>error</i> yang ada mudah dipahami?				
5	Apakah kekontrasan warna huruf sudah sesuai?				
6	Apakah ukuran huruf sudah sesuai?				
7	Apakah jenis huruf sudah sesuai?				
8	Apakah letak tombol pada program sudah sesuai?				
9	Apakah ukuran jendela program sudah sesuai?				
10	Apakah ukuran tombol sudah sesuai?				
11	Apakah keterangan pada tombol mudah dipahami?				
12	Apakah tata letak menu yang ada mudah dipahami?				
13	Apakah tabel penjadwalan mudah dipahami?				
14	Apakah menu-menu yang ada mudah diingat kembali?				
15	Apakah menu-menu yang ada dapat mudah diakses?				
16	Apakah Anda dapat memilih tampilan paket kelas tanpa kesulitan?				
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?				
18	Apakah tombol-tombol yang ada mudah digunakan?				
19	Apakah menu bantuan yang ada cukup membantu?				
20	Apakah combobox penggantian kelas mudah digunakan?				
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?				

Butir-butir pertanyaan di atas telah divalidasi oleh tiga orang ahli materi, yaitu Bapak Dr. Eko Marpanaji (dosen pembimbing), Ibu Kuswari Hernawati, M.Kom dan Bapak Adi Dewanto, M.Kom melalui metode *expert*

judgement. Menurut Sugiyono (2010), metode *expert judgement* digunakan untuk menilai validasi konstruk. Dalam hal ini, setelah instrumen dikonstruksi dari aspek-aspek indikator berdasarkan landasan teori tertentu, maka selanjutnya dikonsultasikan dengan ahli. Hasil validasi dari ahli ini meliputi peminimalan istilah ambigu dan penyempurnaan butir pertanyaan.

Adapun untuk reliabilitas instrumen diuji melalui *software* SPSS dengan teknik Alfa Cronbach dapat dilihat pada gambar 6 sebagai berikut:



Gambar 6. Penghitungan Reliabilitas Instrumen.

Hasil penghitungan reliabilitas dengan teknik Alfa Cronbach pada gambar 7 sebagai berikut:

Scale: ALL

Case Processing Summary			
		N	%
Cases	Valid	3	100.0
	Excluded ^a	0	.0
	Total	3	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics	
Cronbach's Alpha	N of Items
.820	21

Gambar 7. Hasil Penghitungan Reliabilitas

Dari hasil penghitungan didapatkan koefisien reliabilitas Alfa Cronbach sebesar 0.820. Mengacu pada koefisien yang dianggap reliabel yaitu harus lebih besar dari 0,6, maka instrumen variabel *usability* tersebut sudah reliabel.

d. Instrumen untuk Varibel *Portability*

Instrumen untuk varibel *portability* adalah menggunakan media *test case* yang sama dengan instrumen untuk *functionally*. Hanya saja media yang dilakukan untuk alat uji adalah berupa percobaan *software* pada *environmental* yang berbeda. Dalam hal ini, uji coba untuk beberapa sistem operasi komputer. Adapun sistem operasi yang digunakan adalah Windows 7, Linux Mint, Linux Ubuntu, Mac Lion, dan Mac Mountain Lion.

E. Teknik Analisis Data

1. Analisis Kualitas *Correctness*

Faktor kualitas *correctness* dianalisis dengan menghitung jumlah *error* tiap *kilo lines of code (KLOC)*. Jumlah *lines of code* dapat dihitung secara manual melalui fasilitas yang ada dalam aplikasi Netbeans 7.0 karena banyaknya baris data relatif tidak banyak. Sedangkan jumlah *error* dalam pada baris kode pemrograman Java dapat dihitung dengan FindBugs, aplikasi yang dikembangkan oleh *The University of Maryland*. Jumlah *error / KLOC* yang didapatkan dalam pengujian kemudian dibandingkan dengan standar *error / KLOC* pada *industry average* dan standar *Microsoft Application* sesuai dengan yang dikemukakan Anne Mette Jonassen Hass (2008).

2. Analisis Kualitas *Functionally*

Pengujian faktor kaulitas *functionally* dilakukan dengan melakukan tes pada setiap fungsi perangkat lunak. Tes yang dilakukan didokumentasikan dalam *test case*. Setiap *test case* menggambarkan apakah suatu fungsi berjalan sebagaimana mestinya atau tidak.

Berkaitan dengan standar yang digunakan dalam menentukan apakah perangkat lunak telah memenuhi syarat faktor kualitas *functionally*, peneliti menggunakan standar *functionality* yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo*. Standar tersebut ada pada tabel 11 di bawah ini:

Tabel 11. Kriteria Kelolosan Variabel *Functionally*

Kriteria Lolos	Kriteria Gagal
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya.
2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

3. Analisis Kualitas *Usability*

Analisis faktor kualitas *usability* dilakukan dengan metode kuesioner yang akan disebarakan kepada personal terkait penjadwalan di sekolah di SMA N 1 Kalasan. Pengambilan sampel dilakukan dengan teknik *purposive* sampling. Teknik *porposive* sampling yaitu teknik penentuan sampel dengan pertimbangan tententu (Sugiyono, 2010). Dalam hal ini, responden untuk kuesioner adalah tim pembuat jadwal dan guru TIK dalam lingkup sekolah.

Data yang dihasilkan dari kuesioner tersebut merupakan gambaran pendapat atau presepsi pengguna aplikasi penjadwalan. Dalam hal ini yang berkaitan dengan faktor kualitas *usability* perangkat lunak yang dikembangkan. Data yang bersifat kuantitatif tersebut dapat dikonversi ke dalam data kualitatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Menurut Sugiyono (2009), Skala Likert digunakan untuk mengukur sikap, pendapat atau presepsi seseorang atau kelompok terhadap sesuatu, dalam hal

pendapat pengguna terhadap perangkat lunak yang dikembangkan. Data hasil kuesioner yang berupa jawaban – jawaban pengguna dari setiap item dalam kuesioner mempunyai gradasi nilai dari sangat positif sampai sangat negatif. Data tersebut diberi skor seperti pada tabel 12 sebagai berikut:

Tabel 12. Data Skor untuk Kualitas *Usability*.

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Skor yang didapatkan pada tiap hasil kuesioner tersebut kemudian diambil nilai rata - rata. Nilai rata – rata tersebut kemudian dijumlahkan. Dengan jumlah responden sebanyak 3 orang maka dapat dihitung nilai tertinggi dan nilai terendah sebagai berikut:

Nilai tertinggi = $3 \times 21 \times 5 = 315$, dengan asumsi semua responden memberi jawaban sangat setuju pada setiap item kuesioner.

Nilai terendah = $3 \times 21 \times 1 = 63$, dengan asumsi semua responden memberi jawaban “sangat tidak setuju” pada setiap item kuesioner.

Dari data tersebut, kemudian dapat disusun kategori penilaian kuesioner berdasarkan perhitungan interval kelas.

Menghitung Rentang Data

$$\begin{aligned}
 \text{Rentang Data} &= (\text{Data terbesar} - \text{Data terkecil}) + 1 \\
 &= 315 - 63
 \end{aligned}$$

$$= 252$$

Menhitung Panjang Kelas

Panjang Kelas = Rentang Data / Jumlah Kelas

$$= 252 / 5$$

$$= 50.4$$

Dengan data tersebut, kemudian disusun kategori penilain faktor kualitas *usability* berdasarkan interval nilai kuesioner. Adapun interval tersebut ada pada tabel 13 di bawah ini:

Tabel 13. Tabel Interval Nilai Kualitas *Usability*.

Interval Nilai	Kategori
63 – 113	Sangat Tidak Baik
114 – 163	Tidak Baik
164 – 213	Cukup
214 – 263	Baik
264 – 315	Sangat Baik

Jumlah nilai yang didapat dari hasil konversi jawaban kuisisioner ke dalam nilai kuantitatif kemudian dibandingkan dengan kategori penilaian tersebut.

4. Analisis Kualitas *Portability*

Pengujian faktor kualitas *portability* pada penelitian difokuskan untuk menjawab pertanyaan apakah perangkat lunak yang dikembangkan dapat berjalan sebagaimana mestinya pada sistem yang berbeda – beda, dalam hal ini komputer dengan sistem operasi yang berbeda – beda. Sistem operasi yang digunakan adalah Windows 7, Linux Mint, Linux Ubuntu, Mac Lion, dan Mac Mountain Lion.

Berdasarkan acuan bahwa bahasa pemrograman Java merupakan bahasa pemrograman dengan tingkat *portability* yang baik, peneliti menyusun standar pada perangkat lunak yang dikembangkan untuk menentukan apakah perangkat lunak yang dikembangkan lolos atau gagal dalam pengujian faktor kualitas *portability*. Kriteria kelolosan ada pada tabel 14 sebagai berikut:

Tabel 14. Kriteria Kelolosan Analisis *Portability*.

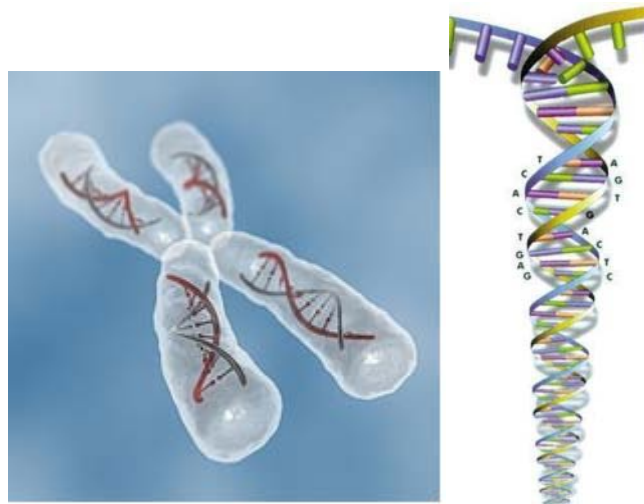
Kriteria Lolos	Kriteria Gagal
Perangkat lunak dapat berjalan sebagaimana mestinya pada setiap sistem yang diujikan.	Paling tidak ada satu sistem dimana perangkat lunak tidak dapat berjalan sebagaimana mestinya.

BAB IV

PEMBAHASAN PENELITIAN

A. Analisis Kebutuhan

SMA N 1 Kalasan sebagai objek observasi jadwal pelajaran dijadikan model permasalahan dalam metode pembuatan algoritma penjadwalan. Adapun model struktur penjadwalan dapat dilihat pada gambar 8 sebagai berikut:



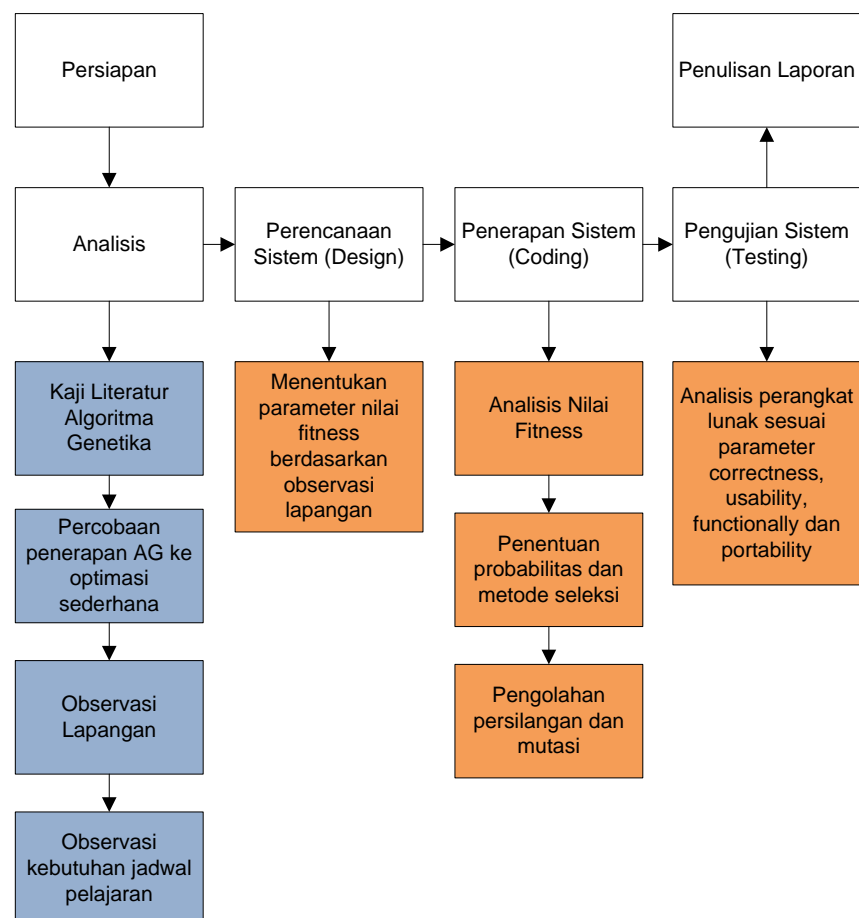
Gambar 8. Kromosom

Gambar di atas adalah bentuk sebuah kromosom yang memiliki susunan gen. Kromosom tersebut diambil mekanismenya ke dalam sebuah paket jadwal. Satu buah paket jadwal diserupakan sebuah kromosom dengan bentuk gen sebagai slot-slot dalam jadwal. Slot-slot dalam jadwal digambarkan pada tabel 15 berikut:

Tabel 15. Contoh Skema Jadwal

Pertemuan	Kelas							
	X1	X2	X3	X4	XI IPA 1	XI IPA 2	...	dst
Senin1	Gen1	Gen2	Gen3					
Senin2								
Senin3								
Senin4								
Selasa1								
Selasa2								
Selasa3								
...
Sabtu4								

B. Desain Penelitian



Gambar 9. Desain Penelitian

Desain penelitian dikembangkan dari model pengembangan *software* linear dengan penambahan sub tahap seperti yang tertampil pada gambar 9. Kesemua tahap tersebut dikembangkan dan dijabarkan pada BAB IV dalam laporan penelitian.

C. Perancangan Penelitian

1. Mekanisme Penjadwalan

Hasil penerapan dan perancangan untuk setiap bagian sistem Algoritma Genetika Penjadwalan Mata Pelajaran Sekolah adalah sebagai berikut:

a. Pembangkitan Individu

Langkah paling awal dalam menentukan jalannya Algoritma Genetika adalah membangkitkan Individu. Individu dalam kasus penjadwalan adalah istilah untuk satu buah entitas penuh sebuah jadwal dalam satu minggu siklus jadwal. Di dalam setiap individu ini, terdapat komponen gen yang menyusun sebuah kesatuan individu. Gen-gen ini diletakkan pada kotak-kotak *array* pada skema jadwal. Komponen gen merupakan representasi dari kode-kode pertemuan, kode guru atau kode mata pelajaran dalam sebuah jadwal. Contoh tabel jadwal dapat dilihat pada tabel 16 berikut:

Tabel 16. Contoh Tabel Jadwal.

Jam Pelajaran	X1	X2	X3	X4
Senin1	Fis1		Mat1	
Senin2	Mat1	Fis1		
Senin3		Mat1	Fis1	
Senin4				
Selasa1				
Selasa2				
Selasa3				
Selasa4				
Rabu1				
Rabu2				
Rabu3				
Rabu4				
Kamis1				
Kamis2				
Kamis3				
Kamis4				
Jumat1				
Jumat2				
Sabtu1				
Sabtu2				
Jumlah jam pelajaran.	20	20	20	20

Tabel di atas adalah contoh sebuah individu jadwal dengan komponen gen sebanyak empat buah. Contoh bentuk jadwal satu individu utuh yang direpresentasikan dalam bentuk lain non tabel adalah seperti pada tabel 17 berikut:

Tabel 17. Contoh Penjadwalan Digambarkan dengan Kode Mapel

0 ---> 7 , 8 , 18 , 13 ,
0 ---> 6 , 15 , 19 , 12 ,
0 ---> 8 , 3 , 10 , 15 ,
0 ---> 16 , 5 , 4 , 16 ,
0 ---> 19 , 7 , 7 , 4 ,
0 ---> 18 , 2 , 12 , 0 ,
0 ---> 14 , 0 , 5 , 9 ,
0 ---> 2 , 18 , 8 , 14 ,
0 ---> 1 , 16 , 0 , 10 ,
0 ---> 9 , 9 , 14 , 6 ,
0 ---> 17 , 6 , 6 , 3 ,
0 ---> 5 , 14 , 16 , 17 ,
0 ---> 15 , 17 , 17 , 19 ,
0 ---> 10 , 1 , 9 , 11 ,
0 ---> 12 , 12 , 3 , 2 ,
0 ---> 3 , 11 , 11 , 8 ,
0 ---> 4 , 13 , 15 , 18 ,
0 ---> 13 , 19 , 13 , 5 ,
0 ---> 0 , 10 , 1 , 1 ,
0 ---> 11 , 4 , 2 , 7 ,

Tabel di atas adalah representasi indeks individu ke 0 beserta gen-gen yang ditulis dalam bentuk angka 0 – 20. Angka ini ditentukan dari jumlah kemungkinan yang boleh muncul dalam susunan gen. Contoh penjadwalan yang digambarkan dengan nama mapel tertampil pada tabel 18 berikut:

Tabel 18. Contoh Penjadwalan Digambarkan dengan Nama Mapel

Senin Jam 1-2 --- > Ekonomi , Olahraga , Bahasa Indonesia 2 , Bahasa Jawa ,
Jam 3-4 --- > Geografi , BK , Bahasa Inggris 2 , Kesenian ,
Jam 5-6 --- > Olahraga , Kimia , Sosiologi , BK ,
Jam 7-8 --- > Pendidikan Karakter , Bahasa Inggris , Bahasa Indonesia 1 , Pendidikan Karakter ,
Selasa Jam 1-2 --- > Bahasa Inggris 2 , Ekonomi , Ekonomi , Bahasa Indonesia 1 ,
Jam 3-4 --- > Bahasa Indonesia 2 , Biologi , Kesenian , Matematika 1 ,
Jam 5-6 --- > Bahasa Asing , Matematika 1 , Bahasa Inggris , Sejarah ,
Jam 7-8 --- > Biologi , Bahasa Indonesia 2 , Olahraga , Bahasa Asing ,
Rabu Jam 1-2 --- > Fisika , Pendidikan Karakter , Matematika 1 , Sosiologi ,
Jam 3-4 --- > Sejarah , Sejarah , Bahasa Asing , Geografi ,
Jam 5-6 --- > Matematika 2 , Geografi , Geografi , Kimia ,
Jam 7-8 --- > Bahasa Inggris , Bahasa Asing , Pendidikan Karakter , Matematika 2 ,
Kamis Jam 1-2 --- > BK , Matematika 2 , Matematika 2 , Bahasa Inggris 2 ,
Jam 3-4 --- > Sosiologi , Fisika , Sejarah , TIK ,
Jam 5-6 --- > Kesenian , Kesenian , Kimia , Biologi ,
Jam 7-8 --- > Kimia , TIK , TIK , Olahraga ,
Jumat Jam 1-2 --- > Bahasa Indonesia 1 , Bahasa Jawa , BK , Bahasa Indonesia 2 ,
Jam 3-4 --- > Bahasa Jawa , Bahasa Inggris 2 , Bahasa Jawa , Bahasa Inggris ,
Sabtu Jam 1-2 --- > Matematika 1 , Sosiologi , Fisika , Fisika ,
Jam 3-4 --- > TIK , Bahasa Indonesia 1 , Biologi , Ekonomi ,

Gen-gen yang ditulis dalam tabel sebelum ini adalah gen yang ditulis dalam bentuk angka. Untuk tabel di atas, gen-gen tersebut dialihtuliskan menjadi nama-nama mata pelajaran. Misal kode gen angka 7 artinya adalah mata pelajaran Ekonomi, maka slot angka 7 pada setiap susunan gen tersebut diganti menjadi mata pelajaran Ekonomi.

Jadwal Pelajaran Sekolah menggunakan Algoritma Genetika, by @andhikalady

Hasil Optimasi Penjadwalan Sekolah dengan Algoritma Genetika

Kelas X

Hari / Jam	Kelas X 1	Kelas X 2	Kelas X 3	Kelas X 4
Senin Jam 1-2	Sosiologi	Ekonomi	Sosiologi	Sejarah
Jam 3-4	Pendidikan Karakter	Bahasa Asing	Sejarah	Bahasa Indonesia 1
Jam 5-6	Fisika	Matematika 1	Bahasa Asing	Bahasa Jawa
Jam 7-8	Olahraga	Matematika 2	Fisika	Matematika 1
Selasa Jam 1-2	Kesenian	Bahasa Indonesia 2	Bahasa Jawa	Ekonomi
Jam 3-4	Sejarah	Sejarah	Kesenian	Biologi
Jam 5-6	Bahasa Jawa	Bahasa Jawa	BK	Fisika
Jam 7-8	Bahasa Inggris 2	Kimia	Ekonomi	Sosiologi
Rabu Jam 1-2	Bahasa Asing	BK	Bahasa Indonesia 2	Bahasa Asing
Jam 3-4	Bahasa Inggris	Kesenian	TIK	Bahasa Indonesia 2
Jam 5-6	Bahasa Indonesia 2	Pendidikan Karakter	Matematika 2	Geografi
Jam 7-8	Bahasa Indonesia 1	Bahasa Inggris	Geografi	Matematika 2
Kamis Jam 1-2	Matematika 2	Geografi	Bahasa Inggris 2	BK
Jam 3-4	Matematika 1	TIK	Bahasa Inggris	Kimia
Jam 5-6	Geografi	Fisika	Olahraga	Bahasa Inggris
Jam 7-8	TIK	Sosiologi	Pendidikan Karakter	Olahraga
Jumat Jam 1-2	BK	Olahraga	Bahasa Indonesia 1	TIK
Jam 3-4	Biologi	Bahasa Inggris 2	Biologi	Kesenian
Sabtu Jam 1-2	Kimia	Biologi	Kimia	Bahasa Inggris 2
Jam 3-4	Ekonomi	Bahasa Indonesia 1	Matematika 1	Pendidikan Karakter

Nilai Error = 6.0
 Nilai Fitness = 166.66666666666666
 Andhika Lady Maharsi. 08520241041

Gambar 10. Hasil *output* Tabel Jadwal

Gambar 10 adalah bentuk tampilan individu satu skema jadwal yang direpresentasikan dalam bentuk GUI Java. Tampilan inilah yang akan muncul ketika program dijalankan.

b. Kode Guru

Kode guru merupakan bentuk representasi *array* gen-gen dalam jadwal. Kode ini memiliki kemiripan langsung dengan kode mapel. Kode guru ini dibuat untuk pengisi slot-slot *array* pada skema jadwal. Jumlah kode guru ini ditentukan berdasarkan jumlah total mata pelajaran atau guru yang ada dalam kasus penjadwalan yang perlu diselesaikan. Jadi semisal jumlah guru dalam sebuah sekolah berjumlah 20 orang. Maka kode gurunya berbentuk seperti berikut:

1. 1 = Mat1 = Matematika 1 = (nama guru Matematika 1)

2. 2 = Mat2 = Matematika 2 = (nama guru Matematika 2)
3. ...
4. Dst, sampai 20.

c. Istilah dalam Penjadwalan

Dalam menyusun rancangan Algoritma Penjadwalan, terdapat beberapa istilah yang dipakai:

- 1) **Kelas angkatan** = sejumlah kelas paralel dalam satu angkatan. Misal kelas X, XI, XII
- 2) **Kelas** = bagian dari kelas angkatan. Misal X1, X2, X3, XI IPA1, dst.
- 3) **Tatap muka / pertemuan** = tatap muka pelajaran dalam seminggu.
- 4) **Jam pelajaran** = bagian dari tatap muka. Dalam sistem yang sudah ada, DUA jam pelajaran dianggap sebagai satu kali tatap muka. Tapi dalam sistem Algoritma Penjadwalan yang peneliti buat, peneliti jarang memakai istilah jam pelajaran, lebih fokus ke tatap muka.
- 5) **Paket tatap muka seminggu** = adalah paket jadwal mata pelajaran seminggu dalam satu kelas.

d. Parameter Jadwal dalam Penelitian

Parameter jadwal adalah batasan dan mekanisme yang menunjukkan bagaimana jadwal tersebut dibuat. Adapun poin-poin parameter adalah sebagai berikut:

- 1) Guru dipisahkan berdasarkan kelas. Jadi guru kelas X tidak mengajar di kelas XI atau XII, begitu pula sebaliknya. Pada bagian ini sengaja dipakai untuk meminimalisir nilai *error*, jadi jumlah guru sengaja diperbanyak.

- 2) Jumlah tatap muka pelajaran siswa adalah 20 kali tatap muka dalam seminggu.
- 3) Hari Senin hingga Kamis terdapat 4 tatap muka. Hari Jumat dan Sabtu masing-masing 2 tatap muka.
- 4) Proses *running* Algoritma Genetika dibuat untuk satu kelas angkatan dahulu, kemudian mengganti kode guru individu terbaik sebanyak 3 kali. Masing-masing untuk kelas X, XI dan XII.
- 5) Jumlah jam mengajar guru adalah 8 pertemuan per minggu.
- 6) Karena pelajaran masih berbentuk tatap muka, maka terjadinya jadwal yang berupa jam pelajaran ganjil (karena satu tatap muka adalah dua jam pelajaran) ditiadakan terlebih dahulu.

e. Pengalokasian jam Pelajaran

Pengalokasian jam pelajaran dapat dilihat dalam tabel 19 di bawah ini:

Tabel 19. Pengalokasian Jam Pelajaran

No	Daftar Mapel	Jumlah jam kelas dalam seminggu	Dikali 4 kelas (karena jumlah kelas ada 4 setiap angkatan)	Jumlah guru pengampu	Jumlah jam per guru	Kode Guru
1	Matematika	2	8	2	4	Mat1,Mat2
2	Fisika	2	8	2	4	Fis1,Fis2
3	Biologi	2	8	2	4	Bio1,Bio2
4	Kimia	2	8	2	4	Kim1,Kim2
5	Bahasa Indonesia	2	8	2	4	Ind1,ind2
6	Bahasa Inggris	2	8	2	4	Ing1,Ing2
7	Geografi	1	4	1	4	Geo1
8	Ekonomi	0	0	0	0	
9	Sosiologi	0	0	0	0	
10	Sejarah	1	4	1	4	Sej1
11	Olahraga	1	4	1	4	Or1
12	TIK	1	4	1	4	TIK1
13	Kesenian	1	4	1	4	Seni1
14	Bahasa Jawa	1	4	1	4	Jaw1
15	Bahasa Asing	1	4	1	4	Asg1
16	BK	1	4	1	4	BK1
17	Agama	0	0	0	0	Agm1
	Total jam pelajaran	20	80	20		

2. Mekanisme Algoritma Genetika

a. Fungsi Objektif dan Penghitungan Nilai *Fitness*.

Nilai *fitness* merepresentasikan tingkat kebugaran sebuah individu. Semakin tinggi nilai *fitness*, maka semakin tinggi pula tingkat kebugaran dan probabilitas untuk lolos seleksi. Nilai *fitness* dihitung berdasarkan tingkat *error* pada setiap individu. Tingkat *error* ini dicari dengan formula fungsi objektif. Formula ini akan menghitung jumlah *error* dalam individu dan mengkalkulasikannya dengan nilai *fitness*. Fungsi objektif dan tingkat *error* pada sistem Algoritma Genetika penjadwalan ditentukan dari parameter sebagai berikut:

- 1) Terjadinya tabrakan jadwal guru dalam kelas angkatan dalam satu tatap muka.
- 2) Terjadinya kemunculan kode guru (atau kemunculan mata pelajaran) lebih dari satu kali dalam satu paket tatap muka seminggu.
- 3) Mata pelajaran olahraga berada dalam tatap muka lebih dari jam ke 2.

Semakin banyak terdapat kejadian a,b dan c pada susunan gen, maka nilai *error* semakin tinggi. Proses penghitungan nilai *error* dapat dijelaskan sebagai berikut seperti pada contoh tabel 20:

- 1) Jumlah tabrakan per hari tatap muka. Satu tabrakan dihitung +1.
- 2) Jumlah kemunculan kode guru dalam satu paket jadwal kelas. Satu kemunculan dobel dihitung +1
- 3) Adanya mapel olahraga pada jam di luar pagi (jam tatap muka 1 dan 2). Satu kemunculan dihitung +1.
- 4) Dengan adanya penambahan nilai *error*, diinginkan kualitas setiap individu dapat dimonitor dan diukur.

Tabel 20. Parameter Penghitungan Nilai *Error*

20 --- >	11	,	9	,	11	,	6	,	>>	dihitung <i>error</i> , muncul dua kali dalam satu deret pertemuan
20 --- >	6	,	18	,	8	,	4	,		
20 --- >	7	,	7	,	9	,	7	,	>>	ini juga termasuk <i>error</i>
20 --- >	9	,	8	,	2	,	6	,		
20 --- >	0	,	0	,	13	,	2	,		
20 --- >	13	,	7	,	3	,	14	,		
20 --- >	8	,	1	,	13	,	18	,		
20 --- >	12	,	3	,	10	,	10	,		
20 --- >	1	,	2	,	11	,	8	,		
20 --- >	19	,	12	,	15	,	19	,		
20 --- >	12	,	16	,	5	,	10	,		
20 --- >	5	,	2	,	14	,	15	,		
20 --- >	15	,	16	,	19	,	12	,		
20 --- >	0	,	4	,	17	,	5	,		
20 --- >	19	,	17	,	18	,	0	,		
20 --- >	15	,	13	,	17	,	1	,		
20 --- >	6	,	11	,	5	,	4	,		
20 --- >	9	,	4	,	18	,	17	,		
20 --- >	16	,	16	,	3	,	14	,		
20 --- >	10	,	3	,	14	,	1	,	>>	angka 15 dan 1 dihitung <i>error</i> karena muncul dua kali dalam satu deret paket jadwal

Setelah penghitungan nilai *error*, yang berikut dilakukan adalah menghitung nilai *fitness*. Nilai *fitness* adalah harga kebugaran pada setiap individu. Semakin tinggi nilai *fitness*, maka individu itu semakin bagus. Jika pada kasus penjadwalan ini kebugaran individu ditentukan dengan banyaknya nilai *error*, maka penghitungan rumus nilai *fitness* adalah sebagai berikut:

$$\text{Nilai Fitness} = \frac{1}{\text{Nilai Error}} \times 1.000$$

Keterangan:

- 1) Angka 1 dijadikan pembagi nilai *error* karena untuk menyebutkan bahwa semakin rendah nilai *error*, maka semakin tinggi nilai *fitness* individu yang bersangkutan.
- 2) Bilangan 1000 sebagai pengali digunakan untuk membuat nilai *fitness* menjadi nilai yang besar dan bulat.

b. Seleksi *Roulette*

Tujuan dari proses seleksi adalah menyaring beberapa individu yang memenuhi syarat nilai *fitness* optimal untuk dilanjutkan proses Algoritma Genetika selanjutnya. Ada beberapa metode seleksi yang dapat digunakan dalam Algoritma Genetika. Contoh yang paling sederhana adalah mengurutkan individu dari *fitness*

besar ke *fitness* kecil, dan kemudian diambil beberapa populasi individu dengan *fitness* paling besar. Tapi metode seperti ini memiliki kelemahan, yakni varietas individu yang terpilih tidak terjaga alias seleksi mati. Untuk itulah proses seleksi yang dilakukan dalam Algoritma Genetika penjadwalan adalah menggunakan model Seleksi *Roulette*. Proses seleksi menggunakan *Roulette Wheel* memungkinkan adanya varietas individu yang tetap terjaga.

Proses dalam melakukan seleksi *Roulette Wheel* adalah sebagai berikut:

1) Bangkitkan probabilitas.

Nilai probabilitas adalah angka yang dikembangkan dari nilai *fitness* pada tiap individu yang digunakan untuk penentuan peluang kelolosan dalam seleksi. Makin tinggi nilai *fitness*, makin tinggi pula probabilitas atau peluang untuk lolos seleksi. Adapun rumus untuk menentukan nilai probabilitas untuk setiap individu adalah sebagai berikut:

$$\text{Nilai Probabilitas} = \frac{\text{nilai fitness}}{\text{total nilai fitness semua individu}} \times 1000$$

Dengan formula seperti di atas, maka setiap individu memiliki kisaran probabilitas tertentu sesuai dengan nilai *fitness*. Contoh hasil penghitungan nilai probabilitas dapat dilihat pada tabel 21 berikut:

Tabel 21. Contoh Hasil Penghitungan Nilai Probabilitas.

Nilai error individu 0 adalah = 26.0 Fitness = 0.0011890606420927466 Probabilitas = 64.49405519487185 Nilai error individu 1 adalah = 29.0 Fitness = 9.182736455463729E-4 Probabilitas = 49.80670378226559 // bagian yang ditandai merupakan hasil probabilitas individu yang bersangkutan
--

Angka yang ditandai merupakan bilangan probabilitas individu yang bersangkutan. Angka ini merupakan peluang kemunculan individu dalam lolos seleksi.

2) Bangkitkan Kode Probabilitas.

Setelah menentukan nilai probabilitas untuk setiap individu, kemudian dibuat penciptaan kode dari angka 0 hingga 999 (seribu buah), yang berisi kode-kode individu yang akan diseleksi sejumlah sama dengan probabilitasnya. Jadi, semakin tinggi nilai *fitness*, probabilitasnya semakin banyak dan peluang untuk lolos seleksi semakin tinggi. Bentuk struktur data probabilitas dapat dilihat pada tabel 22 berikut:

Tabel 22. Bentuk Struktur Data Probabilitas.

```
Individu 1 = probabilitas 64
Individu 2 = probabilitas 49
Dst hingga individu 20.

Maka pemberian nilai pada kode 0 - 999 adalah:

Indeks 0 - 64 ditempati individu 1
Indeks 65 hingga (65+49) ditempati individu 2
Dst hingga angka 1000
```

3) Mengacak bilangan 0 hingga 999 kemudian diambil kode mata pelajaran yang terdapat pada bilangan acak tersebut. Dilakukan sebanyak 20 kali.

Dengan pemilihan seleksi ini, diharapkan individu-individu yang lolos seleksi adalah individu yang memiliki probabilitas tinggi. Pada tahap selanjutnya, individu-individu yang lolos tersebut akan dilakukan proses *crossover* dan mutasi.

c. *Crossover*

Crossover adalah bagian dalam Algoritma Genetika yang cara kerjanya adalah menyilangkan individu yang satu dengan individu lainnya. Setiap individu yang memenuhi syarat *Crossover* akan dipotong menjadi dua sesuai kaidah tertentu

dan menghasilkan dua buah himpunan gen yang berdiri sendiri. Setiap himpunan gen tersebut akan dipasangkan dengan himpunan gen dari individu lain. Mekanisme *crossover* Algoritma Genetika pada kasus penjadwalan Mata Pelajaran Sekolah adalah sebagai berikut:

1) *Crossover rate*

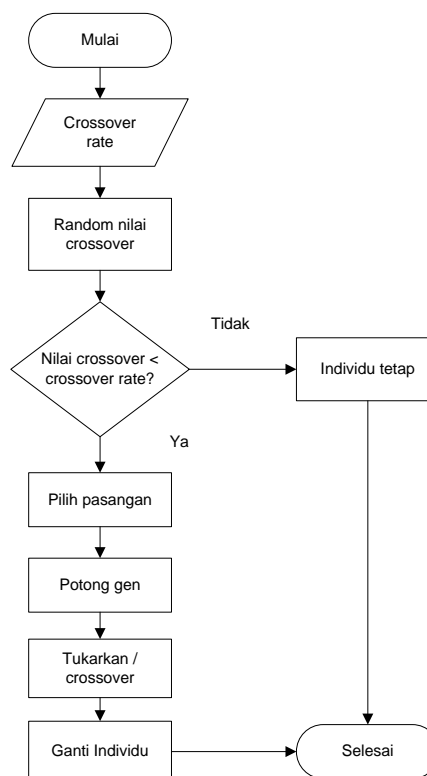
Crossover rate adalah angka yang digunakan sebagai gerbang sebuah Individu apakah memenuhi syarat *crossover* atau tidak. Bilangan *crossover rate* tidak ada aturan tertentu dalam menentukan jumlahnya. Jika nilai *crossover* memenuhi syarat, maka dilakukan *crossover*. Jika tidak, individu yang bersangkutan akan dilewati proses ini.

2) Proses *crossover*

Dalam melakukan proses *crossover*, yang pertama dilakukan adalah membuat pasangan individu. Kaidah dalam menentukan pasangan ini tidak ada pakem khusus. Pasangan individu ini dapat dibuat dari nomor individu ganji-ganjil, genap-genap, atau random sekalipun. Setelah dilakukan pemasangan, yang dilakukan berikutnya adalah membuat *generate crossover rate*. Jika memenuhi syarat, lakukan silang gen. Langkah-langkah *crossover* dapat dijabarkan pada gambar 11 sebagai berikut:

- a) Menentukan *crossover rate*. Misal 20%. Harga *crossover rate* menentukan peluang sebuah individu untuk melakukan *crossover*.
- b) Membagi individu berpasang-pasangan ganjil dan ganjil. Susunannya adalah sebagai berikut: 1 dan 3, 2 dan 4, 5 dan 7, 6 dan 8. Hingga terdapat 10 pasangan *crossover* dari 20 individu yang ada.

- c) Mengacak nomor dari 0 hingga 99 (100 persen) untuk member nilai *crossover rate* pada setiap pasangan. Jika nilai random di bawah *crossover rate*, maka pasangan *crossover* tersebut memenuhi syarat untuk dilakukan *crossover*.
- d) Jika pasangan *crossover* memenuhi syarat di atas, proses berikutnya adalah memotong gen dan memasangkan dengan pasangannya seperti berikut:
- Mengacak nomor dari 0 hingga 4 (4 buah itu ditentukan dari banyaknya kelas) untuk menentukan titik potong gen.
 - Misal angka yang keluar adalah 3. Jadi kedua individu yang berpasangan tersebut dipotong pada kolom ke 3.



Gambar 11. *Flowchart Crossover*

3) Pemrosesan *Crossover* dalam Algoritma Genetika Penjadwalan Sekolah

Jika sepasang individu memenuhi syarat *crossover rate*, maka dilakukan proses *crossover*. Hasil *crossover* di bawah ini adalah model ujicoba *crossover* pada individu 1 (indeks nomor 10) dan individu 2 (indeks nomor 17) dan dipotong pada kolom ke satu. Mekanisme *crossover* dapat dilihat pada tabel 23 berikut:

Tabel 23. Mekanisme *Crossover*.

Individu 1	Individu 2
10 ---> 12, 4, 9, 17, 10 ---> 13, 14, 12, 5, 10 ---> 16, 10, 14, 11, 10 ---> 5, 5, 18, 6, 10 ---> 8, 11, 5, 10, 10 ---> 3, 9, 8, 7, 10 ---> 9, 17, 0, 8, 10 ---> 14, 6, 1, 4, 10 ---> 10, 13, 4, 18, 10 ---> 15, 15, 19, 2, 10 ---> 18, 7, 16, 9, 10 ---> 2, 18, 6, 19, 10 ---> 6, 8, 7, 3, 10 ---> 19, 2, 17, 1, 10 ---> 0, 3, 11, 13, 10 ---> 11, 12, 2, 12, 10 ---> 4, 19, 13, 14, 10 ---> 7, 16, 3, 0, 10 ---> 17, 1, 10, 16, 10 ---> 1, 0, 15, 15,	17 ---> 17, 6, 6, 10, 17 ---> 18, 3, 11, 2, 17 ---> 7, 13, 5, 11, 17 ---> 5, 17, 13, 13, 17 ---> 12, 2, 15, 5, 17 ---> 0, 14, 8, 7, 17 ---> 19, 7, 12, 15, 17 ---> 10, 15, 3, 6, 17 ---> 14, 12, 14, 17, 17 ---> 1, 10, 16, 19, 17 ---> 15, 19, 19, 9, 17 ---> 6, 1, 9, 0, 17 ---> 9, 16, 1, 8, 17 ---> 2, 4, 10, 3, 17 ---> 11, 0, 7, 1, 17 ---> 3, 18, 4, 16, 17 ---> 16, 8, 2, 12, 17 ---> 4, 11, 18, 18, 17 ---> 13, 9, 0, 4, 17 ---> 8, 5, 17, 14,
Individu setelah <i>crossover</i> , dipotong pada kolom pertama.	
1 ---> 12, 6, 6, 10, 2 ---> 13, 3, 11, 2, 3 ---> 16, 13, 5, 11, 4 ---> 5, 17, 13, 13, 5 ---> 8, 2, 15, 5, 6 ---> 3, 14, 8, 7, 7 ---> 9, 7, 12, 15, 8 ---> 14, 15, 3, 6, 9 ---> 10, 12, 14, 17, 10 ---> 15, 10, 16, 19, 11 ---> 18, 19, 19, 9, 12 ---> 2, 1, 9, 0, 13 ---> 6, 16, 1, 8, 14 ---> 19, 4, 10, 3, 15 ---> 0, 0, 7, 1, 16 ---> 11, 18, 4, 16, 17 ---> 4, 8, 2, 12, 18 ---> 7, 11, 18, 18, 19 ---> 17, 9, 0, 4, 20 ---> 1, 5, 17, 14,	

d. Mutasi

Mutasi adalah proses dalam mekanisme Algoritma Genetika yang cara kerjanya adalah mengganti satu atau beberapa gen dari sebuah individu menjadi gen lain yang susunannya random. Pola susunan gen dalam proses mutasi ini tidak ada kaidah tertentu. Karena karakteristik Algoritma Genetika memang disusun dari kerandoman.

1) *Mutation rate*

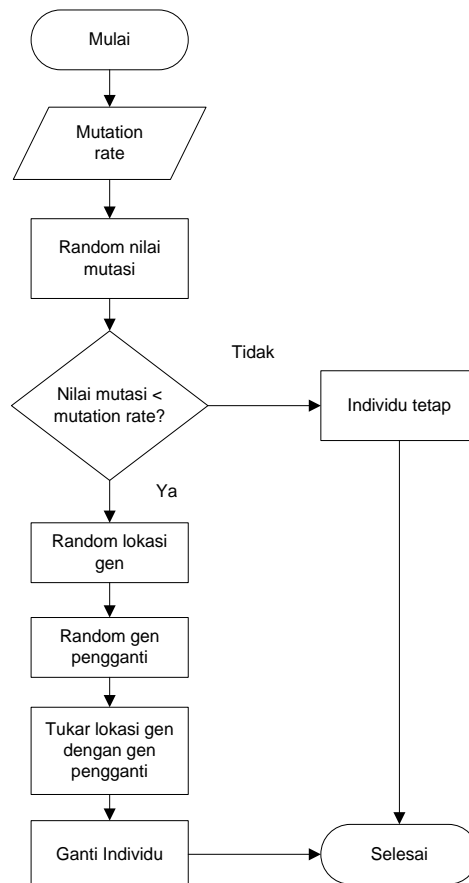
Mutation rate adalah angka yang dijadikan prasyarat dalam menentukan suatu individu dapat dilakukan mutasi atau tidak. Nilai mutasi rate ini tidak memiliki kaidah tertentu.

2) *Proses mutasi*

Proses mutasi hanya dapat dilakukan jika nilai mutasi memenuhi syarat *mutation rate*. Adapun langkah-langkah mutasi terjabar pada gambar 12 dengan proses sebagai berikut:

- a) Menentukan *mutation rate*. Yaitu harga acak yang menentukan probabilitas sebuah individu melakukan mutasi. Harga yang ditentukan dalam aplikasi ini adalah 5 %. Harga ini dapat diubah-ubah.
- b) Karena 5% itu adalah angka yang sedikit, peneliti menggunakan sistem random sebanyak 1000 kali. Dengan estimasi 5% nya adalah 50. Peneliti mengacak sejumlah 1000 kemungkinan dimana sejumlah 50 kemungkinan dapat terjadi mutasi.
- c) Jika hasil pengacakan sebuah individu kurang dari atau sama dengan 50, maka individu tersebut memenuhi syarat untuk dilakukan mutasi.

- d) Jika sebuah individu memenuhi syarat, proses berikutnya adalah mengacak lokasi deret gen. Misal hasil pengacakan adalah deret ke 3, maka perlakukan mutasi dilakukan untuk gen dengan nomor deret ke 3. Deret gen yang mengalami perlakuan mutasi akan dibuang dan diganti dengan deret gen yang baru.



Gambar 12. *Flowchart Mutasi*

3) Pemrosesan Mutasi pada Algoritma Genetika Penjadwalan

Tabel 24 berikut ini adalah contoh hasil mutasi untuk pengacakan kolom mutasi ke 2.

Tabel 24. Pemrosesan Mutasi

Sebelum mutasi	Susunan gen untuk mutasi	Setelah mutasi penggantian kolom gen ke 2
19 --- > 9 , 10 , 3 , 15 ,	19	19 --- > 9 , 19 , 3 , 15 ,
19 --- > 1 , 12 , 10 , 18 ,	18	19 --- > 1 , 18 , 10 , 18 ,
19 --- > 17 , 14 , 8 , 14 ,	5	19 --- > 17 , 5 , 8 , 14 ,
19 --- > 8 , 8 , 14 , 10 ,	17	19 --- > 8 , 17 , 14 , 10 ,
19 --- > 4 , 19 , 15 , 12 ,	0	19 --- > 4 , 0 , 15 , 12 ,
19 --- > 16 , 6 , 16 , 8 ,	14	19 --- > 16 , 14 , 16 , 8 ,
19 --- > 15 , 7 , 2 , 5 ,	9	19 --- > 15 , 9 , 2 , 5 ,
19 --- > 18 , 11 , 17 , 2 ,	7	19 --- > 18 , 7 , 17 , 2 ,
19 --- > 10 , 1 , 1 , 1 ,	13	19 --- > 10 , 13 , 1 , 1 ,
19 --- > 3 , 9 , 0 , 0 ,	8	19 --- > 3 , 8 , 0 , 0 ,
19 --- > 5 , 0 , 18 , 7 ,	10	19 --- > 5 , 10 , 18 , 7 ,
19 --- > 14 , 3 , 11 , 19 ,	2	19 --- > 14 , 2 , 11 , 19 ,
19 --- > 0 , 2 , 19 , 6 ,	16	19 --- > 0 , 16 , 19 , 6 ,
19 --- > 6 , 15 , 12 , 13 ,	12	19 --- > 6 , 12 , 12 , 13 ,
19 --- > 11 , 18 , 9 , 9 ,	4	19 --- > 11 , 4 , 9 , 9 ,
19 --- > 19 , 17 , 4 , 11 ,	6	19 --- > 19 , 6 , 4 , 11 ,
19 --- > 7 , 13 , 13 , 4 ,	11	19 --- > 7 , 11 , 13 , 4 ,
19 --- > 12 , 5 , 7 , 16 ,	15	19 --- > 12 , 15 , 7 , 16 ,
19 --- > 2 , 16 , 6 , 3 ,	3	19 --- > 2 , 3 , 6 , 3 ,
19 --- > 13 , 4 , 5 , 17 ,	1	19 --- > 13 , 1 , 5 , 17 ,

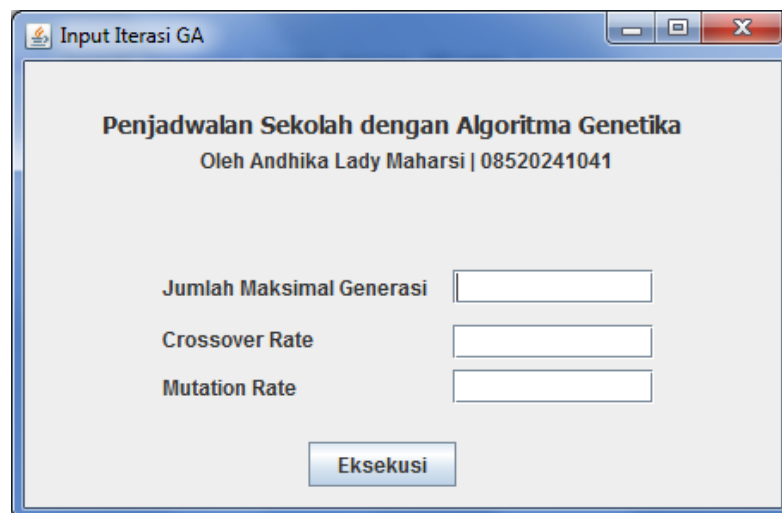
D. Penerapan Sistem Algoritma Genetika terhadap Desain

1. Desain Antarmuka

Desain antarmuka (*interface*) adalah bentuk implemenetasi GUI (*Graphic User Interface*) yang dibuat dengan tools Java untuk memudahkan penggunaan dan pembacaan program. Desain antarmuka ini terdiri dari beberapa jendela yang menyatakan isi program.

a. Input

Input adalah jendela GUI yang pertama kali muncul ketika program dijalankan. Di dalamnya berisi *input* untuk memasukkan bilangan maksimal generasi, *crossover rate* dan *mutation rate*. Gambar 13 dan 12 menunjukkan jendela input.



Gambar 13. Desain GUI untuk Menu *Input*



Gambar 14. Desain GUI untuk Menu Eksekusi

b. Jendela Hasil Penjadwalan

Jendela Hasil Penjadwalan adalah antarmuka yang muncul ketika keseluruhan jadwal telah mencapai nilai optimal untuk ditampilkan. Di dalamnya berisi aturan untuk mengubah tampilan kelas dan aturan untuk *export* XLS hasil penjadwalan. Gambar 15 dan 16 menunjukkan jendela hasil penjadwalan.

Jadwal Pelajaran Sekolah menggunakan Algoritma Genetika, by @andhikalady

Hasil Optimasi Penjadwalan Sekolah dengan Algoritma Genetika

Kelas X Tampil Export XLS Jadwal Kelas X

Hari / Jam	Kelompok 1	Kelompok 2	Kelompok 3	Kelompok 4
Senin Jam 1-2	Pendidikan Karakter	Fisika	Bahasa Indonesia 2	Sejarah
Jam 3-4	BK	Matematika 2	Sejarah	Bahasa Indonesia 2
Jam 5-6	Bahasa Indonesia 2	Ekonomi	Fisika	Bahasa Indonesia 1
Jam 7-8	Fisika	Bahasa Jawa	Olahraga	Matematika 1
Selasa Jam 1-2	Kimia	Matematika 1	Bahasa Indonesia 1	Olahraga
Jam 3-4	Bahasa Inggris 2	Bahasa Inggris 2	TIK	Bahasa Inggris
Jam 5-6	Ekonomi	Kesenian	Biologi	Bahasa Asing
Jam 7-8	Biologi	TIK	Geografi	Fisika
Rabu Jam 1-2	Kesenian	Bahasa Inggris	Kesenian	Bahasa Jawa
Jam 3-4	Geografi	Bahasa Indonesia 1	Ekonomi	TIK
Jam 5-6	Sejarah	Kimia	Matematika 1	Ekonomi
Jam 7-8	Sosiologi	Pendidikan Karakter	Matematika 2	Biologi
Kamis Jam 1-2	Bahasa Inggris	Olahraga	Sosiologi	BK
Jam 3-4	Bahasa Jawa	Bahasa Indonesia 2	Pendidikan Karakter	Geografi
Jam 5-6	Bahasa Asing	Sejarah	BK	Kimia
Jam 7-8	Matematika 1	Biologi	Kimia	Bahasa Inggris 2
Jumat Jam 1-2	Bahasa Indonesia 1	Bahasa Asing	Bahasa Inggris	Sosiologi
Jam 3-4	Matematika 2	Geografi	Bahasa Jawa	Pendidikan Karakter
Sabtu Jam 1-2	TIK	BK	Bahasa Asing	Kesenian
Jam 3-4	Olahraga	Sosiologi	Bahasa Inggris 2	Matematika 2

Nilai Error = 0.0
Nilai Fitness = Infinity

Andhika Lady Maharsi. 08520241041

Gambar 15. Desain GUI untuk Tabel Jadwal

K25 f_x

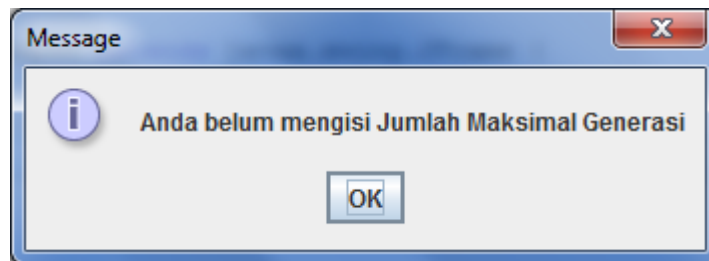
	A	B	C	D	E	F
1	Jadwal	X1	X2	X3	X4	
2	Senin Jam 1-2 --- >	Pendidikan Karakter	Fisika	Bahasa Indonesia 2	Sejarah	
3	Jam 3-4 --- >	BK	Matematika 2	Sejarah	Bahasa Indonesia 2	
4	Jam 5-6 --- >	Bahasa Indonesia 2	Ekonomi	Fisika	Bahasa Indonesia 1	
5	Jam 7-8 --- >	Fisika	Bahasa Jawa	Olahraga	Matematika 1	
6	Selasa Jam 1-2 --- >	Kimia	Matematika 1	Bahasa Indonesia 1	Olahraga	
7	Jam 3-4 --- >	Bahasa Inggris 2	Bahasa Inggris 2	TIK	Bahasa Inggris	
8	Jam 5-6 --- >	Ekonomi	Kesenian	Biologi	Bahasa Asing	
9	Jam 7-8 --- >	Biologi	TIK	Geografi	Fisika	
10	Rabu Jam 1-2 --- >	Kesenian	Bahasa Inggris	Kesenian	Bahasa Jawa	
11	Jam 3-4 --- >	Geografi	Bahasa Indonesia 1	Ekonomi	TIK	
12	Jam 5-6 --- >	Sejarah	Kimia	Matematika 1	Ekonomi	
13	Jam 7-8 --- >	Sosiologi	Pendidikan Karakter	Matematika 2	Biologi	
14	Kamis Jam 1-2 --- >	Bahasa Inggris	Olahraga	Sosiologi	BK	
15	Jam 3-4 --- >	Bahasa Jawa	Bahasa Indonesia 2	Pendidikan Karakter	Geografi	
16	Jam 5-6 --- >	Bahasa Asing	Sejarah	BK	Kimia	
17	Jam 7-8 --- >	Matematika 1	Biologi	Kimia	Bahasa Inggris 2	
18	Jumat Jam 1-2 --- >	Bahasa Indonesia 1	Bahasa Asing	Bahasa Inggris	Sosiologi	
19	Jam 3-4 --- >	Matematika 2	Geografi	Bahasa Jawa	Pendidikan Karakter	
20	Sabtu Jam 1-2 --- >	TIK	BK	Bahasa Asing	Kesenian	
21	Jam 3-4 --- >	Olahraga	Sosiologi	Bahasa Inggris 2	Matematika 2	

Gambar 16. Desain GUI untuk hasil Export XLS

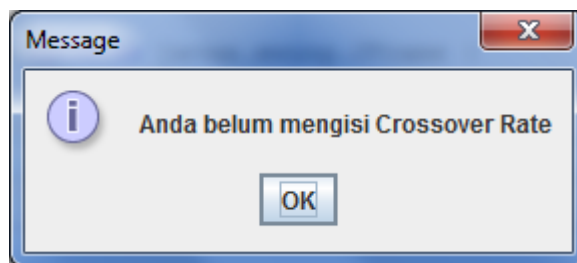
c. Warning

Warning adalah pesan yang menunjukkan kesalahan dalam penjalanan program. Secara umum, *warning* digunakan untuk melindungi pengguna dari kesalahpahaman yang dilakukan oleh pengguna sendiri. Seperti adanya kolom *input*

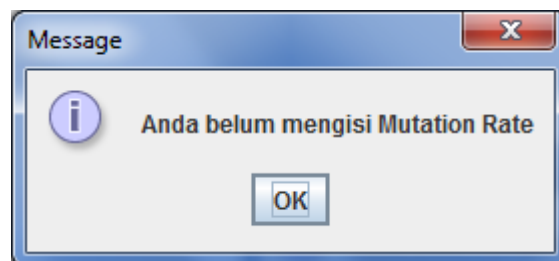
yang belum diisi, variabel *input* yang tidak sesuai, maupun hasil eksekusi yang belum optimal. Gambar 17 s.d. 22 menunjukkan jendela *warning*.



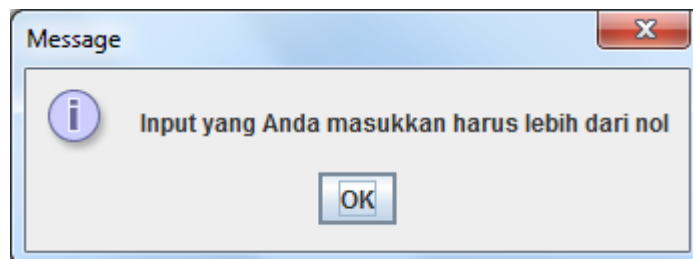
Gambar 17. *Warning* untuk Kesalahan *Input* Maksimal Generasi



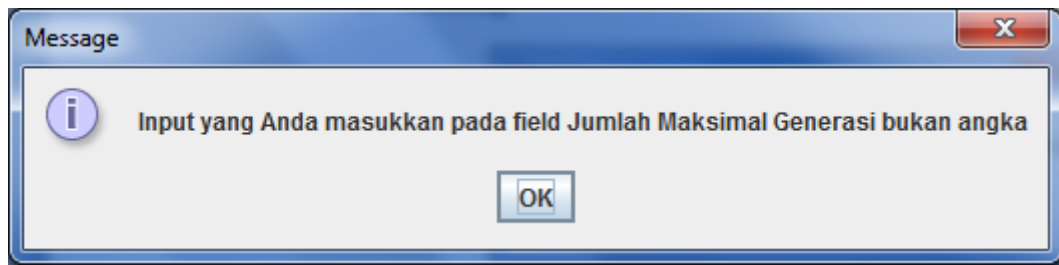
Gambar 18. *Warning* untuk Kesalahan *Input* Crossover.



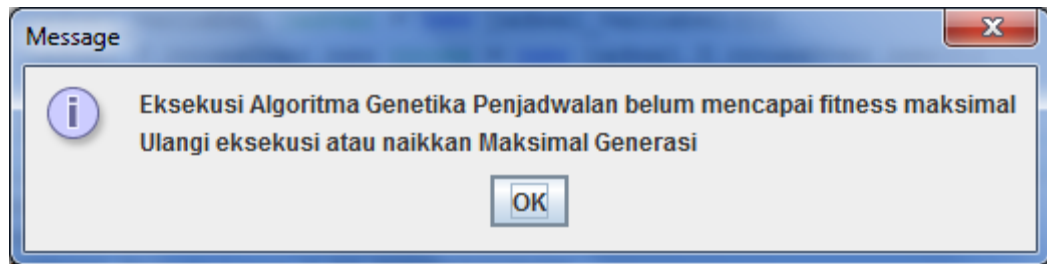
Gambar 19. *Warning* untuk Kesalahan *Input* Mutation Rate



Gambar 20. *Warning* untuk Kesalahan *Input* Bernilai Negatif



Gambar 21. *Warning* untuk Kesalahan *Input* Maksimal Generasi












Gambar 22. *Warning* untuk Belum Mencapai *Fitness* Maksimal.

2. Pengkodean

Pengkodean dilakukan untuk menerjemahkan *flowchart* dan alur penjadwalan menggunakan Algoritma Genetika ke dalam bahasa pemrograman Java. Proses ini memisahkan beberapa alur ke dalam beberapa jenis class program yang berbeda-beda. Hal ini dilakukan supaya dalam proses pengembangan selanjutnya alur program dapat lebih dimengerti dan peluang untuk pengembangan menjadi lebih luas. Tabel 25 merupakan distribusi *class* untuk tiap modul.

Tabel 25. Tabel *Class* untuk *Package* Jadwal Algoritma Genetika

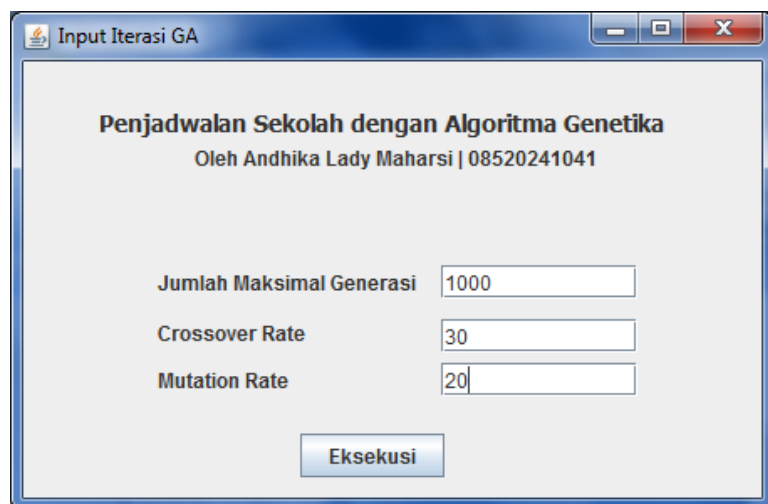
Package / Class	Deskripsi
 Ga.jadwal.source	<i>Package</i> utama sistem penjadwalan mata pelajaran sekolah menggunakan Algoritma Genetika
 Jadwal_variabel	<i>Class</i> yang berisi metode struktur data untuk menyimpan data-data genetika berupa <i>array</i> yang berisi gen-gen.
 Jadwal_1_individu	<i>Class</i> yang berisi <i>method</i> untuk mengeksekusi struktur data pada <i>class</i> Jadwal_variabel dan memanggil <i>method seleksi roulette, crossover</i> dan mutasi. <i>Class</i> ini juga berisi <i>looping</i> yang diiterasikan berdasar angka harapan <i>fitness</i> maksimal.
 Jadwal_2_SeleksiRoulette	<i>Class</i> yang berisi <i>method</i> untuk melakukan seleksi <i>Roulette</i> .
 Jadwal_3_crossover_new	<i>Class</i> yang berisi <i>method</i> untuk mengeksekusi persilangan gen atau <i>crossover</i>
 Jadwal_4_mutasi	<i>Class</i> yang berisi <i>method</i> untuk mengeksekusi proses mutasi genetik.
 Ga.jadwal.gui	<i>Package</i> yang berisi <i>class – class</i> yang menampilkan antar muka (<i>userinterface</i>).
 Input	Antar muka <i>input iterasi</i> .
 TabelJadwalOke	Antar muka <i>Tabel hasil penjadwalan</i> . <i>Class</i> ini membutuhkan tambahan library <i>jxl. 2.6.jar</i> untuk proses export <i>xls</i> .

E. Pengujian Sistem

1. Proses Penjalanan Eksekusi

Untuk menjalankan *software* Penjadwalan langkah-langkah yang harus dilakukan adalah sebagai berikut:

- a. Jalankan program *Software* Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika.
- b. Isikan angka iterasi (maksimal generasi yang diijinkan), *crossover* rate dan *mutation* rate seperti pada gambar 23:



The image shows a software window titled "Input Iterasi GA". Inside the window, the title "Penjadwalan Sekolah dengan Algoritma Genetika" is displayed, followed by the author information "Oleh Andhika Lady Maharsi | 08520241041". Below this, there are three input fields: "Jumlah Maksimal Generasi" with the value "1000", "Crossover Rate" with the value "30", and "Mutation Rate" with the value "20". At the bottom center of the window is a button labeled "Eksekusi".

Gambar 23. *Input* Iterasi

- c. Kemudian tekan tombol Eksekusi.
- d. Setelah itu akan terjadi looping dan eksekusi program berupa seleksi, *crossover*, dan mutasi yang terjadi di belakang layar. Proses ini dapat memakan waktu 0 hingga beberapa detik.
- e. Kemudian jika proses iterasi berhasil mendapat nilai *fitness* maksimal, maka tabel penjadwalan akan muncul seperti gambar 24 berikut:

Jadwal Pelajaran Sekolah menggunakan Algoritma Genetika, by @andhikalady

Hasil Optimasi Penjadwalan Sekolah dengan Algoritma Genetika

Kelas X

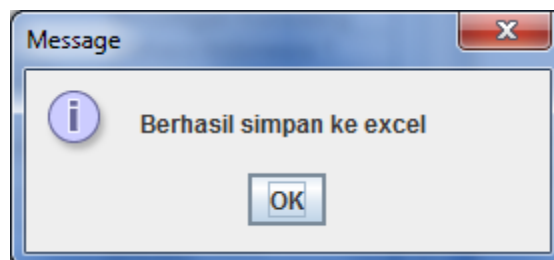
Hari / Jam	Kelompok 1	Kelompok 2	Kelompok 3	Kelompok 4
Senin Jam 1-2	Geografi	BK	Bahasa Asing	Bahasa Inggris
Jam 3-4	Matematika 1	Bahasa Inggris 2	Bahasa Indonesia 2	Kimia
Jam 5-6	Bahasa Inggris 2	Bahasa Indonesia 1	Bahasa Inggris	Biologi
Jam 7-8	Sejarah	Bahasa Asing	Sejarah	Kesenian
Selasa Jam 1-2	TIK	Sosiologi	Bahasa Jawa	Matematika 1
Jam 3-4	Matematika 2	Pendidikan Karakter	Kesenian	Bahasa Jawa
Jam 5-6	Bahasa Inggris	Olahraga	Matematika 1	Geografi
Jam 7-8	Kesenian	Sejarah	Matematika 2	Pendidikan Karakter
Rabu Jam 1-2	Bahasa Indonesia 1	Matematika 1	Kimia	Sosiologi
Jam 3-4	BK	Biologi	Sosiologi	Bahasa Asing
Jam 5-6	Bahasa Indonesia 2	Bahasa Jawa	Fisika	Bahasa Inggris 2
Jam 7-8	Pendidikan Karakter	TIK	Pendidikan Karakter	Olahraga
Kamis Jam 1-2	Kimia	Matematika 2	Olahraga	TIK
Jam 3-4	Biologi	Bahasa Indonesia 2	Ekonomi	BK
Jam 5-6	Bahasa Asing	Bahasa Inggris	Bahasa Inggris 2	Ekonomi
Jam 7-8	Ekonomi	Ekonomi	Geografi	Bahasa Indonesia 2
Jumat Jam 1-2	Sosiologi	Kimia	TIK	Fisika
Jam 3-4	Fisika	Fisika	BK	Matematika 2
Sabtu Jam 1-2	Olahraga	Geografi	Bahasa Indonesia 1	Sejarah
Jam 3-4	Bahasa Jawa	Kesenian	Biologi	Bahasa Indonesia 1

Nilai Error = 0.0
Nilai Fitness = Infinity

Andhika Lady Maharsi. 08520241041

Gambar 24. Hasil Tabel Jadwal

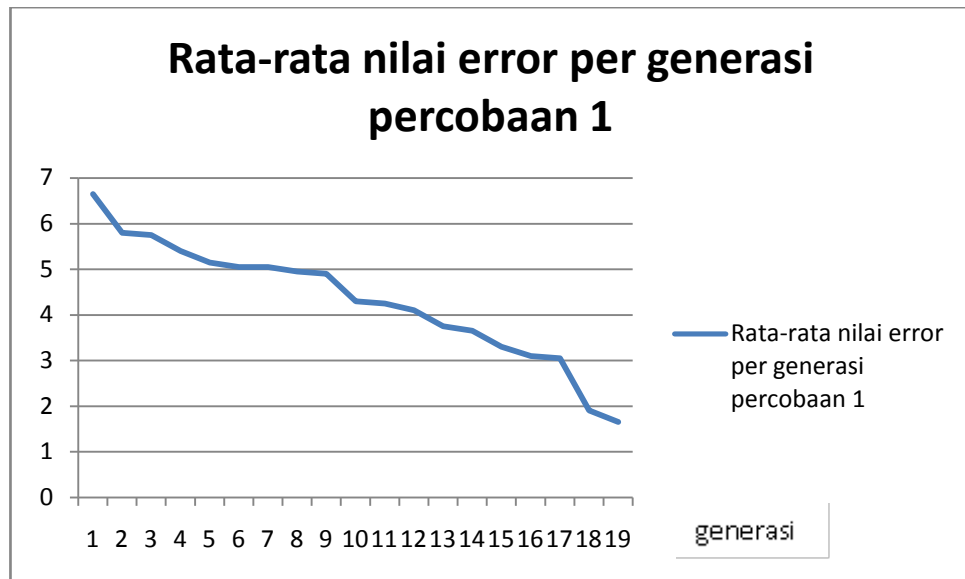
- f. Untuk mengubah format jadwal pelajaran menjadi kelompok IPS atau IPA, pilih choice kelas dan tekan tombol tampil.
- g. Untuk mengekspor tabel jadwal ke dalam bentuk excel atau xls, tekan tombol Export XLS pada jendela hasil optimasi. Jika berhasil, maka akan muncul notifikasi seperti gambar 25 berikut:



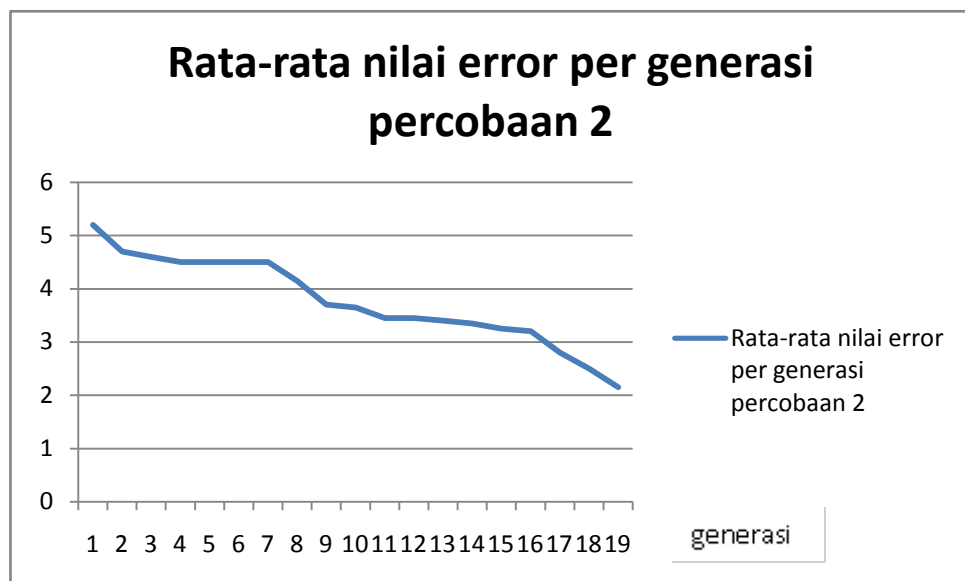
Gambar 25. Warning Keberhasilan Penyimpanan ke xls

2. Hasil Iterasi Nilai Fitness

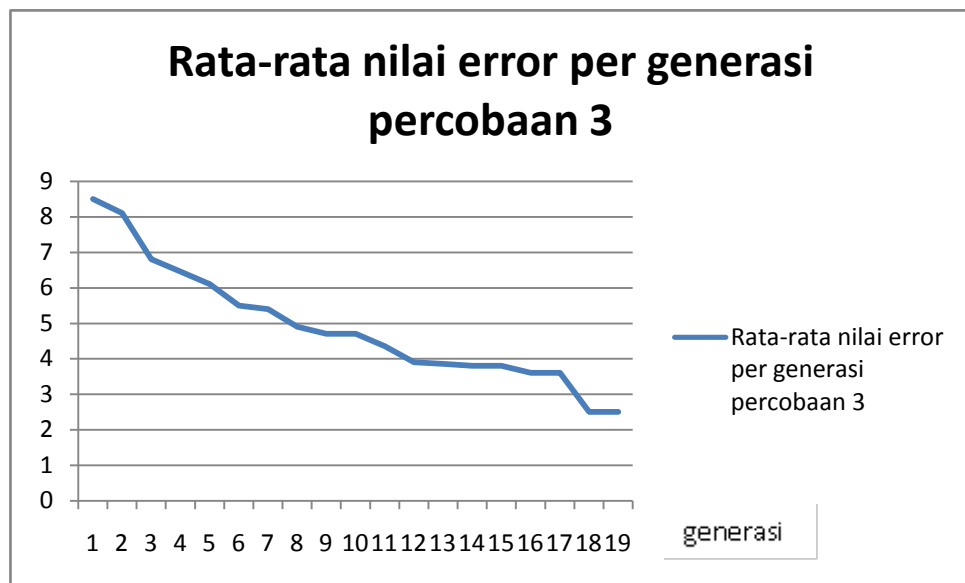
Berdasarkan percobaan hasil eksekusi penghitungan nilai *error* untuk iterasi sebanyak 20 kali, didapatkan pola penurunan rata-rata nilai *error* seperti pada gambar 26 s.d. 30 berikut:



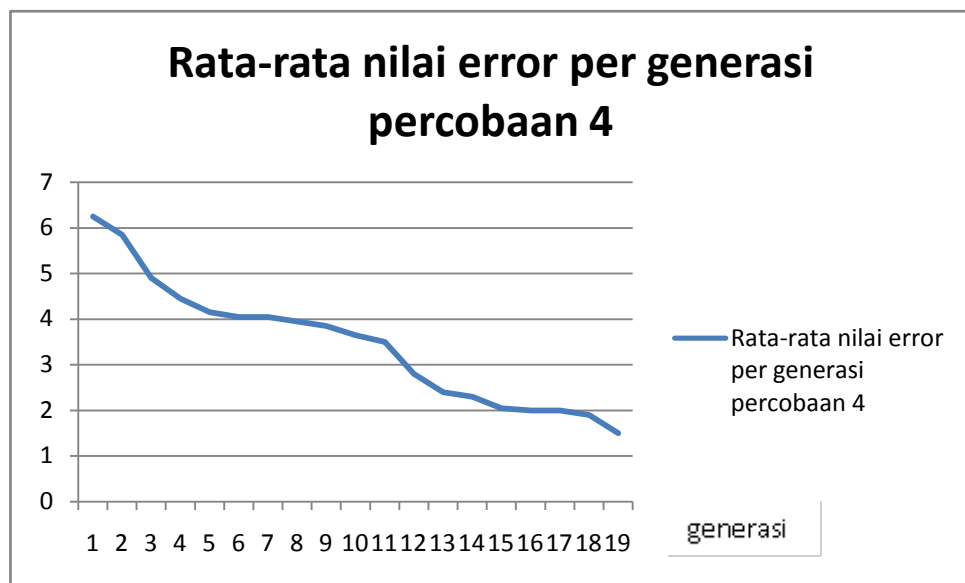
Gambar 26. Grafik Penurunan Nilai *error* Terhadap Iterasi pada Percobaan 1.



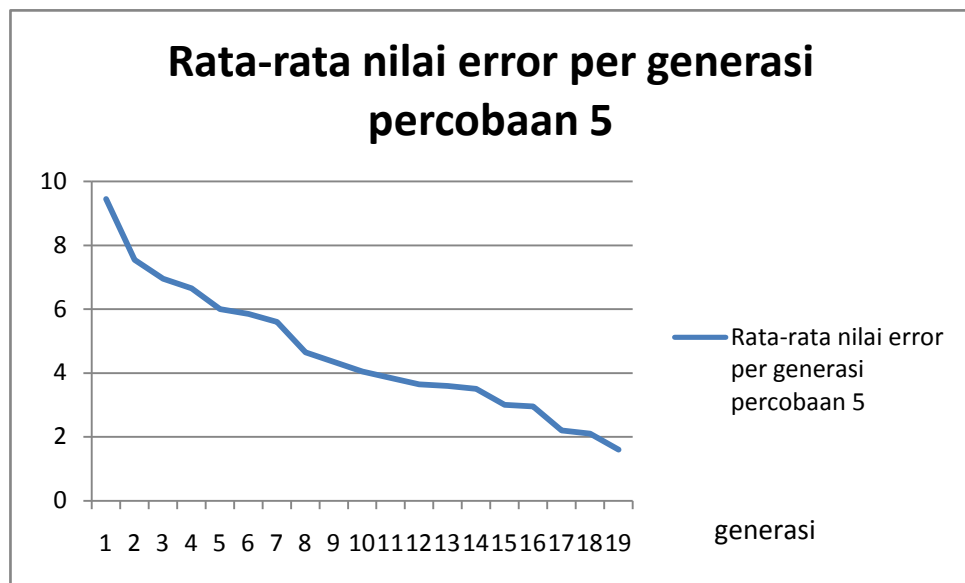
Gambar 27. Grafik Penurunan Nilai *error* Terhadap Iterasi pada Percobaan 2.



Gambar 28. Grafik Penurunan Nilai *error* Terhadap Iterasi pada Percobaan 3.



Gambar 29. Grafik Penurunan Nilai *error* Terhadap Iterasi pada Percobaan 4.



Gambar 30. Grafik Penurunan Nilai *error* Terhadap Iterasi pada Percobaan 5.








3. Analisis dan Testing Algoritma Genetika Penjadwalan

f. Pengujian Kualitas *Correctness*.

1) Menghitung Jumlah Baris Kode

Tabel 26 menampilkan jumlah baris kode per *class*.

Tabel 26. Jumlah Baris Kode

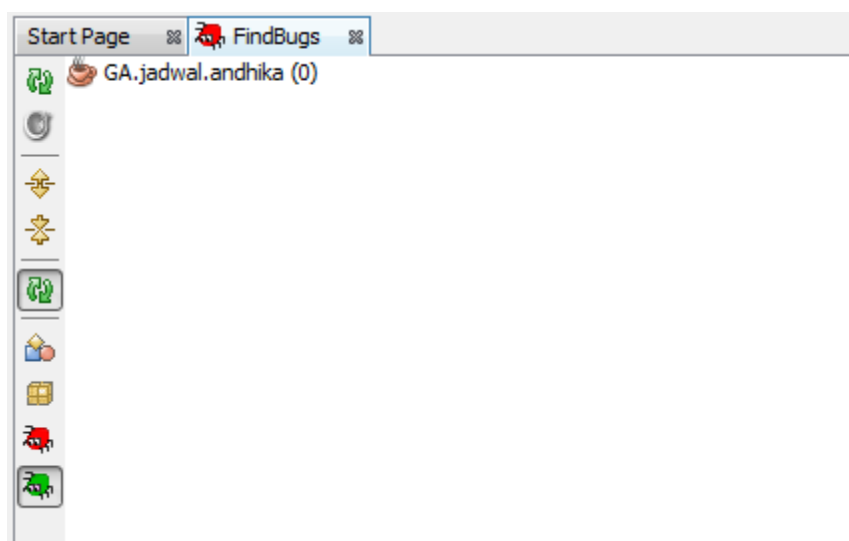
Package / Class	Jumlah Baris Kode
 Jadwal_variabel	296
 Jadwal_1_individu	248
 Jadwal_2_SeleksiRoulette	98
 Jadwal_3_crossover_new	54
 Jadwal_4_mutasi	49
 Input	184
 TabelJadwalOke	377

Jumlah	1306
--------	------

Berdasarkan tabel di atas didapatkan jumlah baris kode sejumlah 1306 LOC (*Lines of Code*) atau 1.31 KLOC (*Kilo Lines of Code*)

2) Menghitung Total *Bugs*

Total *bugs* dihitung menggunakan *tools* Findbugs yang diinstal bersama Netbeans. Findbugs akan menampilkan jumlah bugs yang dihitung berdasarkan baris kode. Dalam proses penghitungan ini, dinyatakan bahwa total bugs pada kode Algoritma Genetika berjumlah 0 seperti pada gambar 31 berikut:



Gambar 31. Hasil Perhitungan Findbugs.

3) Membandingkan Total bugs dengan Line per Code

Berdasarkan hasil pengujian sebelumnya, didapatkan bahwa nilai *error*/KLOC kode Algoritma Genetika adalah 0 *error*/KLOC. Aplikasi akan dikatakan lolos uji jika jumlah *error* lebih sedikit atau sama dengan standar yang digunakan. Sebaliknya, aplikasi akan dikatakan gagal uji jika jumlah *error* melebihi standar yang digunakan.

Tabel 27. Perbandingan Hasil Pengujian Faktor Kualitas *Correctnes*

Nama Standar	Nilai Standar (Error / KLOC)	Hasil Pengujian Aplikasi	Keterangan
<i>Industry Average</i>	1 – 25	$\frac{0}{1.31} = 0$	LOLOS. Jumlah <i>error</i> lebih sedikit dari standar. Lebih baik.
<i>Microsoft Application</i>	0.5		LOLOS. Jumlah <i>error</i> lebih sedikit dari standar. Lebih baik.

Tabel 27 tersebut menunjukan bahwa Aplikasi Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika memenuhi standar faktor kualitas *correctness* **baik** dari Standar Industry Average maupun Microsoft Application.

g. Pengujian Kualitas *Functionally*.

4) Mengelola *Test Case Functionally*.

Pengelolaan *test case functionally* adalah kegiatan melakukan uji alur-alur yang terdapat dalam rencana *test case functionally* yang telah dibuat. Setiap alur pada *test case* akan dicatat hasil *outputnya* (lolos/tidak lolos) dan dibandingkan dengan acuan standar kelolosan *functionally*. Adapun contoh *test case* adalah seperti pada tabel 28 berikut:

Tabel 28. *Test case Functionally* Modeinput01

Test case id	ModeInput01
Purpose	Menguji kesuksesan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi dengan bilangan yang benar.
Assumptions	Jendela <i>input</i> dapat dijalankan.
Test data	Mengisikan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi.
Steps	Jalankan program <i>Software</i> Penjadwalan Inputkan maksimal generasi, <i>rate crossover</i> , dan <i>rate</i> mutasi berupa angka Klik OK
Expected result:	Eksekusi Algoritma Genetika berjalan.
Actual result:	Eksekusi Algoritma Genetika berjalan.
Pass/Fail:	Lolos

Terdapat sejumlah enam *test case* dengan kasus dan alur berbeda-beda (terlampir). Masing-masing *test case* akan dicatat dalam pengelolaan dan dibandingkan dengan acuan kelolosan *test case*. Adapun hasil pencatatan setiap alur *test case functionally* tertampil pada tabel 29 berikut:

Tabel 29. Hasil Pengelolaan *Test case*

No	Nama Tes	Peneliti	Validator 1	Validator 2
1	ModeInput01	Lolos	Lolos	Lolos
2	ModeInput01	Lolos	Lolos	Lolos
3	ModeRunning01	Lolos	Lolos	Lolos
4	ModeRunning01	Lolos	Lolos	Lolos
5	ModeDokumentasi01	Lolos	Lolos	Lolos
6	ModeDokumentasi01	Lolos	Lolos	Lolos

5) Melakukan Perbandingan Test case dengan Kriteria Kelolosan

Tabel 30 menampilkan kriteria kelolosan, sedangkan tabel 31 menampilkan hasil kelolosan.

Tabel 30. Tabel Kriteria Kelolosan

Kriteria Lolos	Kriteria Gagal
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya.
2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	2. Jika ada fungsi yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

Tabel 31. Tabel Hasil Uji *Functionally*

Hasil Pengujian	Keterangan
Semua fungsi primer dan fungsi pendukung berjalan dengan baik.	Lolos

Berdasarkan nilai kelolosan pada uji *functionally* di tabel, maka dapat disimpulkan bahwa *Software* Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah **lolos** uji *functionally*.

h. Pengujian Kualitas Usability.

6) Mengelola Hasil Angket *Usability*.

Nilai usabilitas Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika dihasilkan sebesar 280 dari total 3 responden yaitu satu orang guru TIK dan dua orang guru petugas pembuat jadwal di kurikulum sekolah. Sistem angket menggunakan skala Likert yang terdiri dari lima persepsi, berurutan dari sangat baik, baik, cukup, kurang dan sangat kurang. Angka 280 tersebut jika dialihkan menjadi bentuk persen adalah sebagai berikut:

$$\text{Presentase kepuasan} = \frac{\text{skor perolehan}}{\text{perolehan maksimal}} \times 100$$

$$\text{Presentase kepuasan} = \frac{280}{315} \times 100$$

$$\text{Presentase kepuasan} = 88,89 \%$$

7) Membandingkan dengan Kategori Angket Skala Likert.

Hal yang berikut dilakukan adalah membandingkan dengan kategori skala Likert. Adapun kategorinya tertampil pada tabel 32 berikut:

Tabel 32. Kategori *Usability* Skala Likert.

Interval Nilai	Kategori
63 – 113	Sangat Tidak Baik
114 – 163	Tidak Baik
164 – 213	Cukup
214 – 263	Baik
264 – 315	Sangat Baik

Hasil angket faktor *usability* adalah didapatkannya nilai usabilitas sebesar **280** atau **88,89 %**. Jika dibandingkan dengan kategori *usability* skala Likert pada poin analisis, angka tersebut termasuk dalam kategori **sangat baik**.

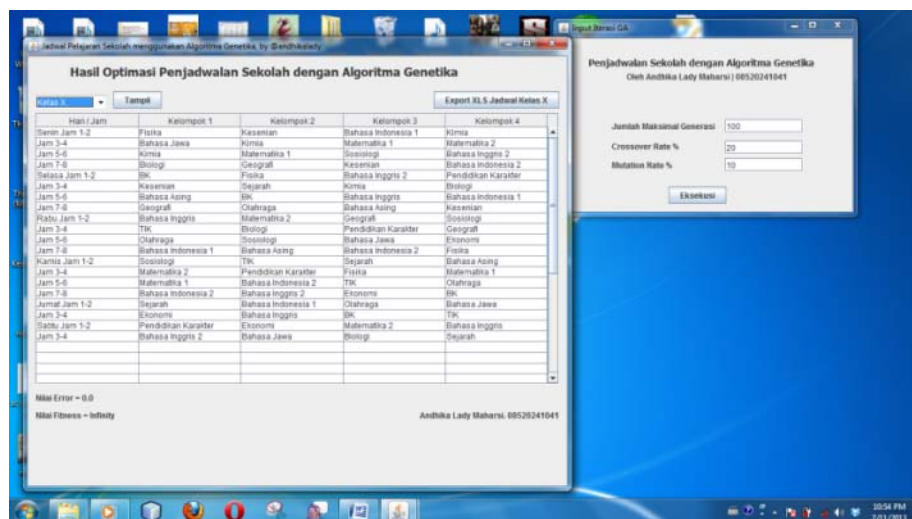
i. Pengujian Kualitas Portability.

8) Mengelola *Test Case Portability*.

Test case portability terdiri dari alur yang berisi *test* untuk pengecekan keberhasilan program dijalankan pada *environmental* yang berbeda, dalam hal ini adalah sistem operasi yang berbeda-beda. Contoh *test case* untuk *portability* sistem operasi Windows 7 adalah seperti pada tabel 33 dan gambar 32 berikut:

Tabel 33. *Test Case Portability* untuk OS Windows 7.

Test case id	ModeWindows
Purpose	Menguji kesuksesan jalannya program di Windows 7
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Windows 7
Actual result:	Program dapat dijalankan di Windows 7
Pass/Fail:	Lolos



Gambar 32. Hasil *Running Test Portability* pada OS Windows 7

Adapun hasil pengelolaan *test case* untuk masing-masing sistem operasi adalah seperti pada tabel 34 berikut:

Tabel 34. Hasil Uji *Portability*

No	Environmental	Peneliti	Validator 1	Validator 2
1	Windows 7	Lolos	Lolos	Lolos
2	Linux Ubuntu	Lolos	Lolos	Lolos
3	Linux Mint 13 Maya	Lolos	Lolos	Lolos
4	Mac OS Mountain Lion	Lolos	Lolos	Lolos
5	Mac OS Lion	Lolos	Lolos	Lolos

Berdasarkan hasil tabel tersebut, dapat disimpulkan bahwa sistem telah lolos dalam berbagai lingkungan yang berbeda-beda, sehingga dapat dikatakan Sistem Penjadwalan Mata Pelajaran menggunakan Algoritma Genetika memiliki kualitas portabilitas yang **baik**.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan uraian rangkaian penelitian Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika maka dapat ditarik kesimpulan sebagai berikut:

1. Pendekatan Algoritma Genetika dalam penyelesaian masalah konstrain penjadwalan sekolah dapat berhasil karena mampu mencari kombinasi penjadwalan yang tepat dengan nilai *fitness* maksimal, nilai *error* minimal (ditemukan nilai *error* 0), dan solusi yang optimal (tidak terjadinya tabrakan jadwal pelajaran dan tidak terjadinya kejadian yang menimbulkan peningkatan nilai *error*).
2. Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah diuji dengan kualitas *correctness* menggunakan kaidah KLOC dan menghasilkan nilai *bugs* sejumlah 0 line(s). Ini membuktikan bahwa Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah memenuhi standar kualitas *correctness*. Kemudian diuji dengan kualitas *functionally* dan menghasilkan catatan alur fungsi primer dan fungsi sekunder telah berjalan dengan baik. Ini membuktikan bahwa aplikasi tersebut telah memenuhi standar kaidah *functionally*. Berdasarkan aspek *usability*, Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah diujikan kepada *user*, dalam hal ini adalah komponen guru atau persona sekolah yang bertugas menyusun jadwal sekolah, telah menghasilkan kesimpulan “sangat

baik” untuk faktor *usability*. Berdasarkan aspek *portability*, Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah diujikan untuk beberapa *environmental* atau sistem operasi komputer yang berbeda-beda dan berhasil berjalan baik di semua kondisi. Hal ini menunjukkan bahwa Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika telah memenuhi standar portabilitas yang baik.

B. Saran

Beberapa saran yang dapat digunakan dalam penelitian ini adalah sebagai berikut:

1. Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika baru memiliki dua *constraint* nilai *error* yaitu tabrakan jadwal dan kemuculan mata pelajaran olahraga di siang hari. Dalam penelitian selanjutnya, diharapkan *constraint* nilai *error* dapat ditambahkan dan ikut dihitung dalam pembobotan nilai *fitness* semisal randomisasi guru, mata pelajaran ganjil, dan randomisasi mata pelajaran.
2. Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika baru memfasilitasi sejumlah empat kelas dan belum memiliki fasilitas untuk *input* manual. Dalam pengembangan selanjutnya diharapkan sudah terdapat fasilitas randomisasi kelas dan *input* manual.
3. Pendekatan Algoritma Genetika dalam pembuatan sistem penjadwalan menggunakan bilangan genetika sebagai berikut: jumlah maksimal generasi 400, *crossover rate* 40%, *mutation rate* 20%, metode seleksi menggunakan *Roulette Wheel*, pengkodean kromosom menggunakan bit kode pelajaran, tidak menggunakan kaidah *elitism*, menggunakan bahasa pemrograman Java,

penempatan slot kromosom diletakkan pada struktur data *array* multidimensi. Dalam penelitian selanjutnya diharapkan dapat dilakukan penelitian lebih lanjut dalam hal perbedaan hasil *output*, kecepatan, efektifitas dan lain-lain jika menggunakan bilangan maksimal generasi dan *rate* yang berbeda-beda. Selain itu, penggunaan bahasa pemrograman lain, struktur data yang dibuat dan metode seleksi juga perlu diteliti lebih lanjut.

4. Penelitian perangkat lunak baru diuji dalam empat aspek yaitu *correctness*, *functionally*, *usability* dan *portability*. Dalam penelitian selanjutnya diharapkan aspek kualitas perangkat lunak yang lain dapat ikut disertakan.

DAFTAR PUSTAKA

- Allen, F. (1999). Using Genetic Algorithm to find Technical Trading Rules. *Journal Financial Economics* .
- Arikunto, P. D. (2010). *Manajemen Penelitian*. Jakarta: Rineka Cipta.
- Bevan, N. (1995). Measuring *Usability* as Quality of Use. *Software Quality Journal* .
- Coley, D. A. (2000). *An Introduction to Genetic Algorithm*. Singapore: World Scientific Publishing Co. Pte. Ltd.
- Fikri, R. (2005). *Pemrograman Java*. Yogyakarta: Penerbit ANDI.
- General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo* (2005)
- Gibbon, M. P. (2004). Sistem Penjadwalan Matakuliah menggunakan Algoritma Genetika (Studi Kasus Fakultas MIPA IPB).
- Guide to Advanced Software Testing*. (2008). Norwood: Artech House
- Java. (2013). *Learn About Java Technology*. Diakses Mei 29, 2013, from Java.com: <http://www.java.com/en/about/>
- Kusumadewi, S. &. (2005). *Penyelesaian Masalah optimasi dengan Teknik-teknok Heuristik*. Yogyakarta: Graha Ilmu.
- Ladjamudin, A. B. (2006). *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- McConnell, S. (2004). *Code Complete*. Redmond: Microsoft Press.
- Mitsuo Gen, Runwei Cheng. (2000). *Genetic Algorithm and Engineering Optimization*. New York: John Willey & Sons, Inc.
- Nielsen, J. (1993). *Usability Engineering*. New York: Morgan Kaufmann.
- Pengenalan Pemrograman Java 1*. (2007). Jakarta: J.E.N.I
- Sam'ani. (2012). *Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika*.
- Saputro, N. (2003). Pengenalan Huruf dengan memakai Algoritma Genetik. *Integral* .
- Software Engineering : A Practitioner's Approach*. (2010). New York: McGraw Hill
- Software Engineering and Testing*. (2010). Sudbury: Jones and Bartlett Publishers

- Sugiyono. (2009). *Metode Penelitian Pendidikan*. Bandung: Alfabeta
- Sugiyono. (2010). *Statistika untuk Penelitian*. Bandung: Alfabeta
- Suyanto. (2005). *Algoritma Genetika dalam Matlab*. Yogyakarta: Andi.
- Syaifullah. (2008). *Trick Menyusun Jadwal Pelajaran*.

LAMPIRAN

Lampiran 1. *Test case Functionally*

Test case id	ModeInput01
Purpose	Menguji kesuksesan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi dengan bilangan yang benar.
Assumptions	Jendela <i>input</i> dapat dijalankan.
Test data	Mengisikan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi.
Steps	<ul style="list-style-type: none"> - Jalankan program <i>Software</i> Penjadwalan - Inputkan maksimal generasi, <i>rate crossover</i>, dan <i>rate</i> mutasi berupa angka - Klik OK
Expected result:	Eksekusi Algoritma Genetika berjalan.
Actual result:	Eksekusi Algoritma Genetika berjalan.
Pass/Fail:	Lolos

Lampiran Tabel 1. *Test case* ModeInput01

Test case id	ModeInput02
Purpose	Menguji kesuksesan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi dengan format yang salah.
Assumptions	Jendela <i>input</i> dapat dijalankan.
Test data	Mengisikan <i>input</i> maksimal generasi, <i>rate crossover</i> dan <i>rate</i> mutasi.
Steps	<ul style="list-style-type: none"> - Jalankan program <i>Software</i> Penjadwalan - Inputkan maksimal generasi, <i>rate crossover</i>, dan <i>rate</i> mutasi berupa bukan angka. - Klik OK
Expected result:	Eksekusi Algoritma Genetika tidak berjalan, muncul peringatan kesalahan <i>input</i> dan permintaan untuk <i>input</i> ulang.
Actual result:	Eksekusi Algoritma Genetika tidak berjalan, muncul peringatan kesalahan <i>input</i> dan permintaan untuk <i>input</i> ulang.
Pass/Fail:	Lolos

Lampiran Tabel 2. *Test case* ModeInput02

Test case id	ModeRunning01
Purpose	Menguji kesuksesan running program setelah <i>input</i> dijalankan.
Assumptions	Sudah menginputkan variabel pada jendela <i>input</i> dengan benar.
Test data	Mengetest kesuksesan <i>running</i> program.
Steps	<ul style="list-style-type: none"> - Inputkan maksimal generasi dengan angka lebih dari 100 (angka minimal 100 digunakan sebagai angka perkiraan eksekusi Algoritma Genetika dapat mencapai nilai <i>fitness</i> maksimal), <i>rate crossover</i> dan <i>rate</i> mutasi. - Klik OK
Expected result:	Eksekusi Algoritma Genetika berjalan, muncul jendela tabel jadwal.
Actual result:	Eksekusi Algoritma Genetika berjalan, muncul jendela tabel jadwal.
Pass/Fail:	Lolos

Lampiran Tabel 3. *Test case* ModeRunning01

Test case id	ModeRunning02
Purpose	Menguji kesuksesan <i>warning</i> pada <i>running</i> program setelah <i>input</i> dijalankan.
Assumptions	Sudah menginputkan variabel pada jendela <i>input</i> dengan benar.
Test data	Mengetest kesuksesan <i>warning</i> pada <i>running</i> program.
Steps	<ul style="list-style-type: none"> - Inputkan maksimal generasi dengan angka kurang dari 10 (angka maksimal 10 digunakan sebagai angka perkiraan eksekusi Algoritma Genetika belum dapat mencapai nilai <i>fitness</i> maksimal), <i>rate crossover</i> dan <i>rate</i> mutasi. - Klik OK
Expected result:	Eksekusi Algoritma Genetika berjalan, muncul pop up peringatan "Eksekusi belum berhasil, ulangi eksekusi atau naikkan nilai Maksimal Generasi"
Actual result:	Eksekusi Algoritma Genetika berjalan, muncul pop up peringatan "Eksekusi belum berhasil, ulangi eksekusi atau naikkan nilai Maksimal Generasi"
Pass/Fail:	Lolos

Lampiran Tabel 4. *Test case* ModeRunning02

Test case id	ModeDokumentasi01
Purpose	Menguji kesuksesan pengubahan tabel jadwal ke kelas X, IPA atau IPS.
Assumptions	Eksekusi Algoritma Genetika telah berhasil memunculkan tabel dan

	mencapai nilai <i>fitness</i> maksimal.
Test data	Mengetest kesuksesan perubahan tabel jadwal ke kelas X, IPA atau IPS.
Steps	<ul style="list-style-type: none"> - Ubah choice kelas ke kelas X, IPA atau IPS - Klik Tampil
Expected result:	<ul style="list-style-type: none"> - Tabel jadwal berubah menjadi <i>choice</i> yang diharapkan. - Button Export Excel berubah text menjadi export <i>choice</i> yang diharapkan.
Actual result:	<ul style="list-style-type: none"> - Tabel jadwal berubah menjadi <i>choice</i> yang diharapkan. - Button Export Excel berubah text menjadi export <i>choice</i> yang diharapkan.
Pass/Fail:	Lolos

Lampiran Tabel 5. *Test case* ModeDokumentasi01

Test case id	ModeDokumentasi02
Purpose	Menguji kesuksesan export XLS jadwal.
Assumptions	Eksekusi Algoritma Genetika telah berhasil memunculkan tabel dan mencapai nilai <i>fitness</i> maksimal.
Test data	Mengetest kesuksesan export XLS jadwal.
Steps	<ul style="list-style-type: none"> - Ubah <i>choice</i> kelas ke IPA / IPS - Klik Tampil - Klik Export XLS
Expected result:	<ul style="list-style-type: none"> - Tabel jadwal berubah menjadi <i>choice</i> yang diharapkan. - Button Export Excel berubah text menjadi export <i>choice</i> yang diharapkan. - Ketika Export XLS muncul peringatan file telah berhasil terexport.
Actual result:	<ul style="list-style-type: none"> - Tabel jadwal berubah menjadi <i>choice</i> yang diharapkan. - Button Export Excel berubah text menjadi export <i>choice</i> yang diharapkan. - Ketika Export XLS muncul peringatan file telah berhasil terexport.
Pass/Fail:	Lolos

Lampiran Tabel 6. *Test case* ModeDokumentasi02

No	Nama Tes	Peneliti	Validator 1	Validator 2
1	ModeInput01	Lolos	Lolos	Lolos
2	ModeInput01	Lolos	Lolos	Lolos
3	ModeRunning01	Lolos	Lolos	Lolos
4	ModeRunning01	Lolos	Lolos	Lolos
5	ModeDokumentasi01	Lolos	Lolos	Lolos
6	ModeDokumentasi01	Lolos	Lolos	Lolos

Lampiran Tabel 7. Hasil Pengelolaan *Test case*

Lampiran 2. Hasil Butir Pertanyaan *Usability*

No	Butir Pertanyaan	Sub Indikator			
		U	L	O	A
1	Apakah kegunaan program dapat dikenali dari awal interface?				
2	Apakah bahasa yang ditampilkan mudah dipahami?				
3	Apakah hasil <i>output</i> tabel penjadwalan mudah dipahami?				
4	Apakah pesan <i>error</i> yang ada mudah dipahami?				
5	Apakah kontras warna huruf sudah sesuai?				
6	Apakah ukuran huruf sudah sesuai?				
7	Apakah jenis huruf sudah sesuai?				
8	Apakah letak tombol pada program sudah sesuai?				
9	Apakah ukuran jendela program sudah sesuai?				
10	Apakah ukuran tombol sudah sesuai?				
11	Apakah keterangan pada tombol mudah dipahami?				
12	Apakah tata letak menu yang ada mudah dipahami?				
13	Apakah tabel penjadwalan mudah dipahami?				
14	Apakah menu-menu yang ada mudah diingat kembali?				
15	Apakah menu-menu yang ada dapat mudah diakses?				
16	Apakah Anda dapat memilih tampilan paket kelas tanpa kesulitan?				
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?				
18	Apakah tombol-tombol yang ada mudah digunakan?				
19	Apakah menu bantuan yang ada cukup membantu?				
20	Apakah combobox penggantian kelas mudah digunakan?				
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?				

Lampiran Tabel 8. Penjabaran Butir Pertanyaan Aspek *Usability*.

Keterangan:

U: Understandability

L : Learnability

O: Operability

A: Attractiveness

Lampiran 3. Hasil Angket *Usability*.

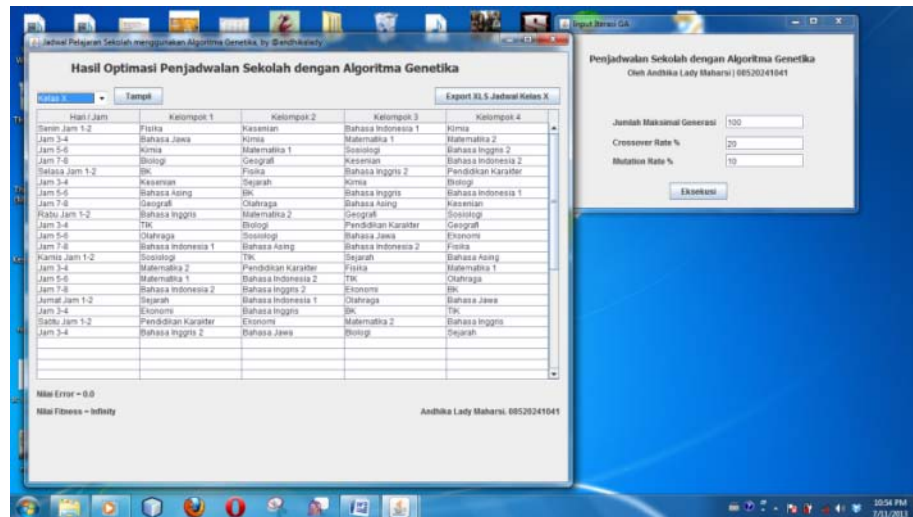
No	Jawaban Angket																					Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Responden 1	4	4	4	3	4	4	4	4	4	4	4	4	4	5	5	5	5	4	4	4	5	88
Responden 2	4	4	5	4	5	5	5	4	4	5	5	5	4	4	4	5	5	5	4	5	5	96
Responden 3	4	4	5	4	5	5	5	5	5	4	4	5	4	4	4	5	5	5	4	5	5	96
Jumlah	12	12	14	11	14	14	14	13	13	13	13	14	12	13	13	15	15	14	12	14	15	280

Lampiran Tabel 9. Hasil Angket *Usability*

Lampiran 4. *Test case Portability*

Test case id	ModeWindows
Purpose	Menguji kesuksesan jalannya program di Windows 7
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Windows 7
Actual result:	Program dapat dijalankan di Windows 7
Pass/Fail:	Lolos

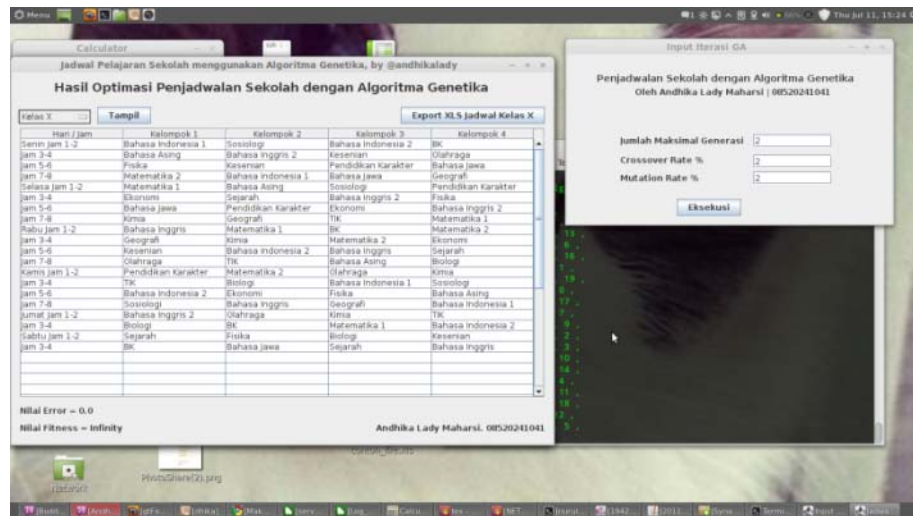
Lampiran Tabel 10. *Test case ModeWindows*



Lampiran Gambar 1. Hasil *Running Test Portability* pada OS Windows 7

Test case id	ModeLinuxMint
Purpose	Menguji kesuksesan jalannya program di Linux Mint 13 Maya
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Linux Mint 13 Maya
Actual result:	Program dapat dijalankan di Linux Mint 13 Maya
Pass/Fail:	Lolos

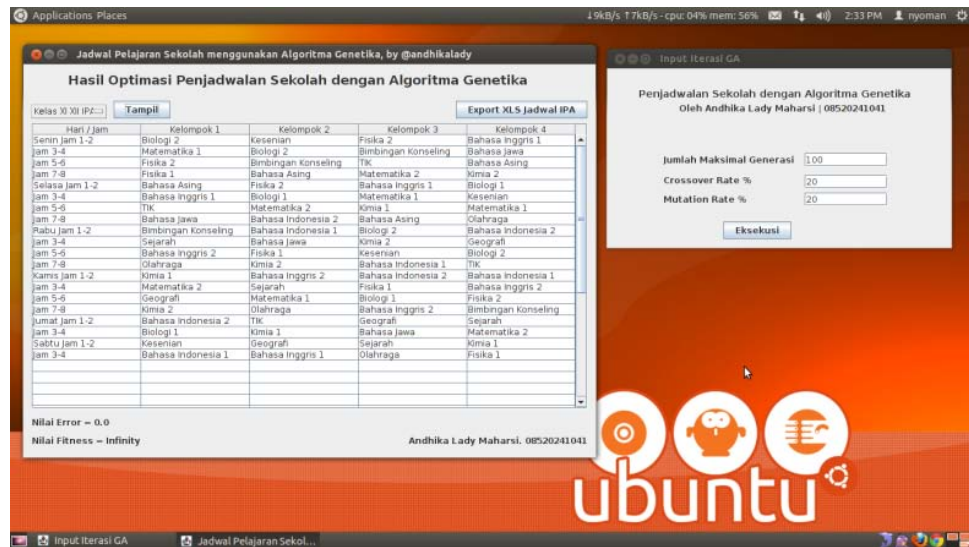
Lampiran Tabel 11. *Test case ModeLinuxMint*



Lampiran Gambar 2. Hasil *Running Test Portability* pada OS Linux Mint 13 Maya

Test case id	ModelinuxUbuntu
Purpose	Menguji kesuksesan jalannya program di Linux Ubuntu.
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Linux Ubuntu.
Actual result:	Program dapat dijalankan di Linux Ubuntu.
Pass/Fail:	Lolos

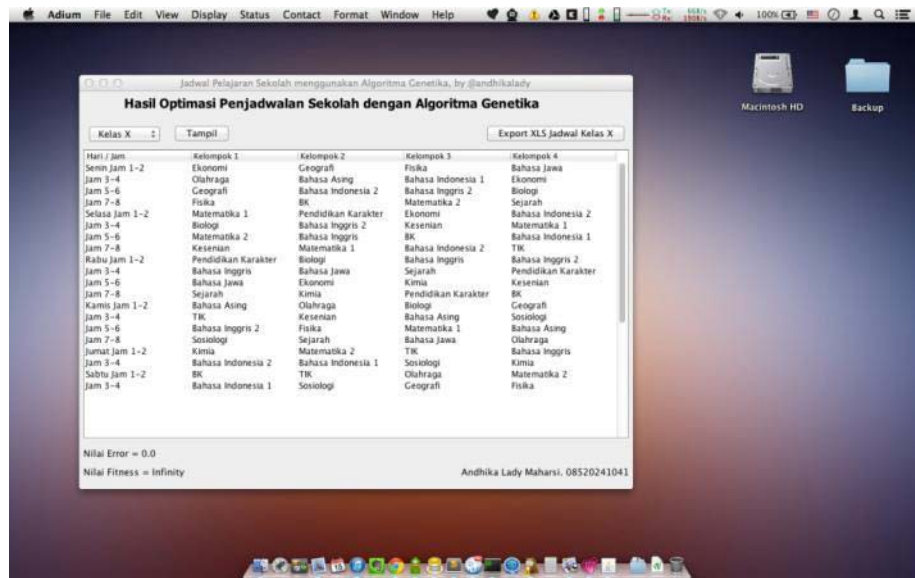
Lampiran Tabel 12. *Test case* ModelinuxUbuntu



Lampiran Gambar 3. Hasil *Running Test Portability* pada OS Linux Ubuntu

Test case id	ModeMacOs Mountain Lion
Purpose	Menguji kesuksesan jalannya program di Mac OS Mountain Lion
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Mac OS Mountain Lion
Actual result:	Program dapat dijalankan di Mac OS Mountain Lion
Pass/Fail:	Lolos

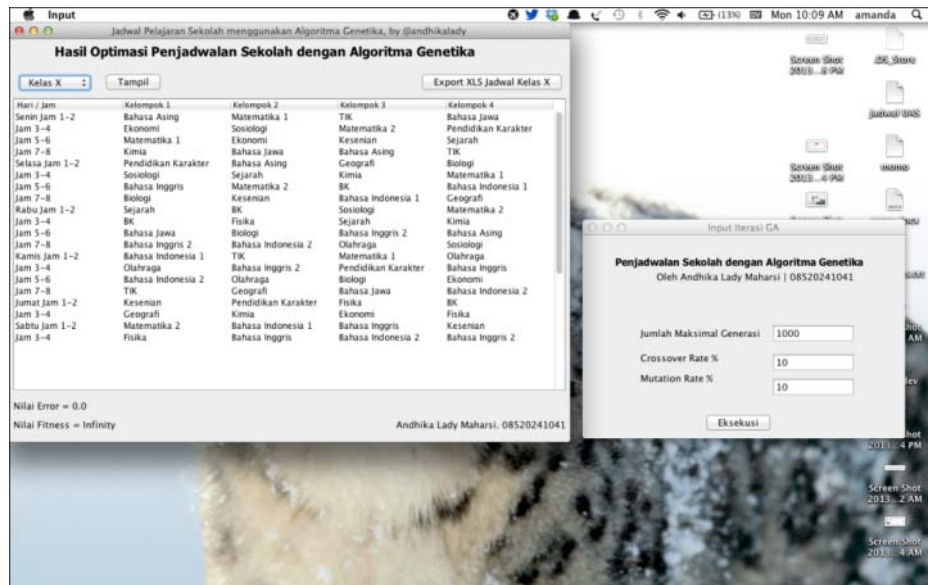
Lampiran Tabel 13. *Test case* ModeMac Mountain Lion



Lampiran Gambar 4. Hasil *Running Test Portability* pada OS MACOS Mountain Lion

Test case id	ModeMacOS Lion
Purpose	Menguji kesuksesan jalannya program di Mac Lion
Assumptions	Komputer sudah terinstal Java.
Test data	Membuka program.
Steps	- Membuka program Ga.jadwal.andhika.jar
Expected result:	Program dapat dijalankan di Mac Lion
Actual result:	Program dapat dijalankan di Mac Lion
Pass/Fail:	Lolos

Lampiran Tabel 14. *Test case* ModeMacOS Lion



Lampiran Gambar 5. Hasil *Running Test Portability* pada OS MACOS Lion

Lampiran 5. Kode Program

Input.java
<pre> package ga.jadwal.gui; import ga.jadwal.code.jadwal_1_individu; import ga.jadwal.code.jadwal_variabel; import javax.swing.JOptionPane; /** * * @author Andhika Lady Maharsi */ public class input extends javax.swing.JFrame { /** * Creates new form input */ jadwal_1_individu jadwal1 = new jadwal_1_individu(); jadwal_variabel jadvar = new jadwal_variabel(); public input() { initComponents(); } @SuppressWarnings("unchecked") // <editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() { jLabel1 = new javax.swing.JLabel(); jLabel2 = new javax.swing.JLabel(); jLabel3 = new javax.swing.JLabel(); jLabel4 = new javax.swing.JLabel(); jTextField2 = new javax.swing.JTextField(); jTextField3 = new javax.swing.JTextField(); jButton1 = new javax.swing.JButton(); jLabel5 = new javax.swing.JLabel(); jTextField4 = new javax.swing.JTextField(); jButton2 = new javax.swing.JButton(); setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE); setTitle("Input Iterasi GA"); jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N jLabel1.setText("Penjadwalan Sekolah dengan Algoritma Genetika"); jLabel2.setText("Jumlah Maksimal Generasi"); </pre>


```
jLabel3.setText("Crossover Rate %");

jLabel4.setText("Mutation Rate %");

jButton1.setText("Eksekusi");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jLabel5.setText("Oleh Andhika Lady Maharsi | 08520241041");

jTextField4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField4ActionPerformed(evt);
    }
});

jButton2.setText("Petunjuk");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(jLabel1)
                        .add(jLabel5)
                    )
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .add(jButton1)
                )
                .add(jLabel4)
            )
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .add(jButton2)
            .addContainerGap())
);

```

```

layout.createSequentialGroup()
    .addGap(80, 80, 80)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(jLabel3)
    .addComponent(jLabel4))
    .addGap(15, 15, 15)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(jTextField3)
    .addComponent(jTextField4)
    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addContainerGap(58, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(28, 28, 28)
        .addComponent(jLabel1)
        .addGap(4, 4, 4)
        .addComponent(jLabel5)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 56,
Short.MAX_VALUE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel3)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel4))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jButton2)
    .addComponent(jButton1))
.addGap(15, 15, 15))
);

pack();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    if (jTextField2.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Anda belum mengisi Jumlah Maksimal
Generasi");
    }
    if (jTextField3.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Anda belum mengisi Crossover Rate");
    }
    if (jTextField4.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Anda belum mengisi Mutation Rate");
    }
    try {
        int inputan = Integer.parseInt(jTextField2.getText());

    } catch (Exception z) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan pada field
Jumlah Maksimal Generasi bukan angka");
        return;
    }
    try {
        int inputan = Integer.parseInt(jTextField3.getText());

    } catch (Exception z) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan pada field
Crossover Rate bukan angka");
        return;
    }
    try {
        int inputan = Integer.parseInt(jTextField4.getText());

    } catch (Exception z) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan pada field
Mutation Rate bukan angka");
        return;
    }
    if (Integer.parseInt(jTextField2.getText()) < 1) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan harus lebih
dari nol");
    }
}

```

```

        return;
    }
    if (Integer.parseInt(jTextField3.getText()) < 1) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan harus lebih
dari nol");
        return;
    }
    if (Integer.parseInt(jTextField4.getText()) < 1) {
        JOptionPane.showMessageDialog(null, "Input yang Anda masukkan harus lebih
dari nol");
        return;
    }
    System.out.println("Isi 2: " + jTextField2.getText().toString());
    System.out.println("Isi 3: " + jTextField3.getText().toString());
    System.out.println("Isi 4: " + jTextField4.getText().toString());

    jadwal1.setLoop(Integer.parseInt(jTextField2.getText().toString()));

    jadvar.inisialisasi();
    jadwal1.inisialisasi();
}

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Petunjuk:\n- Jumlah Maksimal Generasi
adalah iterasi maksimal Algoritma Genetika yang diijinkan.\n- Crossover rate adalah
derajat kemungkinan individu mengalami crossover.\n- Mutation Rate adalah derajat
kemungkinan individu mengalami mutasi.");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new input().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
// End of variables declaration
}

```

Tabeljadwal.java

```

package ga.jadwal.gui;

import ga.jadwal.code.jadwal_1_individu;
import ga.jadwal.code.jadwal_variabel;
import java.io.FileWriter;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.awt.Desktop;
import java.io.File;
import java.io.IOException;

/**
 *
 * @author Andhika Lady Maharsi
 */
public class TabelJadwalOke extends javax.swing.JFrame {

    private jadwal_variabel jadvar1;
    jadwal_variabel jadvar = new jadwal_variabel();

    public TabelJadwalOke() {
        inisialisasi(0);
    }

    public DefaultTableModel getTableModel() {
        return (DefaultTableModel) jTable1.getModel();
    }

    public final void inisialisasi(int j) {
//        jadvar.inisialisasi();
        initComponents();
        pilihan();
        cobatampilTabel(j); //menampilkan tabel dengan indeks no 0 karena indeks 0 adalah
        individu terbaik. Jangan 1 tjoy!
        tampilHari();
    }

    public void pilihan() {

```

```

choice1.add("Kelas X");
choice1.add("Kelas XI XII IPA");
choice1.add("Kelas XI XII IPS");
}

public void tampilToTabel(int sql) {
    kosongkanTabel();
    for (int i = 0; i < jadwal_variabel.pertemuan[sql].length; i++) {
        System.out.print((sql + 1) + " --- > ");
        for (int j = 0; j < jadwal_variabel.pertemuan[sql][i].length; j++) {
            System.out.print(jadwal_variabel.pertemuan[sql][i][j] + " , ");
            getTableModel().setValueAt(jadwal_variabel.pertemuan[sql][i][j], i, j);
        }
        System.out.println("");
    }
    getTableModel().setValueAt(1, 1, 0);    // horeee
}

public void tampilHari() {
    for (int i = 0; i < jadwal_variabel.pertemuan[1].length; i++) {
        addBarisKosong();
        getTableModel().setValueAt(jadwal_variabel.nama_hari[i], i, 0);
    }
}

public void cobatampilTabel(int index) {
    for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {

getTableModel().setValueAt(jadwal_variabel.nama_mapel[jadwal_variabel.pertemuan[in
dex][i][j]], i, j + 1);
//      getTableModel().setValueAt(jadwal_variabel_kelas_X.nama_mapel[i], i, j+1);
        }
    }
    ButtonExportExcel.setText("Export XLS Jadwal Kelas X");
    jLabel3.setText("Nilai error = " + jadwal_1_individu.nilai_error[index]);
    jLabel4.setText("Nilai Fitness = " + jadwal_1_individu.fitness[index]);
}

public void cobatampilTabelIPA(int index) {

    for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {

getTableModel().setValueAt(jadwal_variabel.nama_mapelIPA[jadwal_variabel.pertemua
n[index][i][j]], i, j + 1);
//      getTableModel().setValueAt(jadwal_variabel_kelas_X.nama_mapel[i], i, j+1);
        }
    }
}

```

```

    }
    ButtonExportExcel.setText("Export XLS Jadwal IPA");

    jLabel3.setText("Nilai error = " + jadwal_1_individu.nilai_error[index]);
    jLabel4.setText("Nilai Fitness = " + jadwal_1_individu.fitness[index]);
}

public void cobatampilTabelIPS(int index) {
    for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {

            getTableModel().setValueAt(jadwal_variabel.nama_mapelIPS[jadwal_variabel.pertemuan
[index][i][j]], i, j + 1);
            //      getTableModel().setValueAt(jadwal_variabel_kelas_X.nama_mapel[i], i, j+1);
        }
    }
    ButtonExportExcel.setText("Export XLS Jadwal IPS");
    jLabel3.setText("Nilai error = " + jadwal_1_individu.nilai_error[index]);
    jLabel4.setText("Nilai Fitness = " + jadwal_1_individu.fitness[index]);
}

public void tampilTabelOtomatis(int index) {

    for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {
            //
            getTableModel().setValueAt(jadwal_variabel_kelas_X.nama_mapel[jadwal_variabel_kelas_X.pertemuan[index][i][j]], i, j+1);
            getTableModel().setValueAt(jadwal_variabel.nama_mapel[i], i, j + 1);
        }
    }

    jLabel3.setText("Nilai error = " + 0);
    jLabel4.setText("Nilai Fitness = " + 1000);
}

private void kosongkanTabel() {
    int k = 0;
    while (k < getTableModel().getRowCount()) {
        getTableModel().removeRow(0);
        k++;
    }
}

private void addBarisKosong() {
    getTableModel().addRow(new Object[]{null, null, null, null, null});
}

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    choice1 = new java.awt.Choice();
    ButtonExportExcel = new javax.swing.JButton();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Jadwal Pelajaran Sekolah menggunakan Algoritma Genetika, by
@andhikalady");
    setName("Jadwal Pelajaran Sekolah menggunakan Algoritma Genetika, by
@andhikalady"); // NOI18N

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null}
        },
        new String [] {
            "Hari / Jam", "Kelompok 1", "Kelompok 2", "Kelompok 3", "Kelompok 4"
        }
    ));

```



```
jScrollPane1.setViewportView(jTable1);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel1.setText("Hasil Optimasi Penjadwalan Sekolah dengan Algoritma Genetika");

jLabel2.setText("Andhika Lady Maharsi. 08520241041");

jLabel3.setText("jLabel3");

jLabel4.setText("jLabel4");

ButtonExportExcel.setText("Import XLS");
ButtonExportExcel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ButtonExportExcelActionPerformed(evt);
    }
});

jButton1.setText("Tampil");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jScrollPane1)
                .add(jLabel1)
                .add(ButtonExportExcel)
                .add(jButton1))
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .add(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .add(jLabel3)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .add(jLabel4)
            .addContainerGap())
);
```

```

        .addContainerGap()
        .addComponent(choice1, javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
428, Short.MAX_VALUE)
        .addComponent(ButtonExportExcel)
        .addGroup(layout.createSequentialGroup())
        .addGap(63, 63, 63)
        .addComponent(jLabel1)))
        .addContainerGap()
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addComponent(jLabel1)
        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(choice1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(ButtonExportExcel)
        .addComponent(jButton1)))
        .addGap(7, 7, 7)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 404,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 25,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(11, 11, 11)
    );

    pack();
} // </editor-fold>

private void ButtonExportExcelActionPerformed(java.awt.event.ActionEvent evt) {
    if ("Export XLS Jadwal IPA".equals(ButtonExportExcel.getText())) {
        try {
            FileWriter fw = new FileWriter("D:/file_IPA.csv");

```

```

        fw.write("Jadwal");
        fw.append(",");
        fw.write("XI_IPA1");
        fw.append(",");
        fw.write("XI_IPA2");
        fw.append(",");
        fw.write("XII_IPA1");
        fw.append(",");
        fw.write("XII_IPA2\n");

        for (int i = 0; i < jadwal_variabel.pertemuan[0].length; i++) {
            fw.write(jadwal_variabel.nama_hari[i] + " --- > ");
            fw.append(",");
            for (int j = 0; j < jadwal_variabel.pertemuan[0][i].length; j++) {
fw.write(jadwal_variabel.nama_mapelIPA[jadwal_variabel.pertemuan[0][i][j]]);
                fw.append(",");
            }
            fw.write("\n");
        }

        fw.flush();
        fw.close();
        JOptionPane.showMessageDialog(null, "Berhasil simpan ke excel ");
        File file = new File("D:/file_IPA.csv");
        try {
            Desktop.getDesktop().open(file);
        } catch (IOException e) {
        }
    } catch (Exception e) {
    }
}

else if ("Export XLS Jadwal IPS".equals(ButtonExportExcel.getText())) {
    try {
        FileWriter fw = new FileWriter("D:/file_IPS.csv");
        fw.write("Jadwal");
        fw.append(",");
        fw.write("XI_IPS1");
        fw.append(",");
        fw.write("XI_IPS2");
        fw.append(",");
        fw.write("XII_IPS1");
        fw.append(",");
        fw.write("XII_IPS2\n");

        for (int i = 0; i < jadwal_variabel.pertemuan[0].length; i++) {
            fw.write(jadwal_variabel.nama_hari[i] + " --- > ");

```

```

        fw.append(",");
        for (int j = 0; j < jadwal_variabel.pertemuan[0][i].length; j++) {
fw.write(jadwal_variabel.nama_mapelIPS[jadwal_variabel.pertemuan[0][i][j]]);
            fw.append(",");
        }
        fw.write("\n");
    }

    fw.flush();
    fw.close();
    JOptionPane.showMessageDialog(null, "Berhasil simpan ke excel ");
    File file = new File("D:/file_IPS.csv");
    try {
        Desktop.getDesktop().open(file);
    } catch (IOException e) {
    }
} catch (Exception e) {
}
}

else {
    try {
        FileWriter fw = new FileWriter("D:/file_kelas_X.csv");
        fw.write("Jadwal");
        fw.append(",");
        fw.write("X1");
        fw.append(",");
        fw.write("X2");
        fw.append(",");
        fw.write("X3");
        fw.append(",");
        fw.write("X4\n");

        for (int i = 0; i < jadwal_variabel.pertemuan[0].length; i++) {
            fw.write(jadwal_variabel.nama_hari[i] + " --- > ");
            fw.append(",");
            for (int j = 0; j < jadwal_variabel.pertemuan[0][i].length; j++) {
                fw.write(jadwal_variabel.nama_mapel[jadwal_variabel.pertemuan[0][i][j]]);
                fw.append(",");
            }
            fw.write("\n");
        }

        fw.flush();
        fw.close();
        JOptionPane.showMessageDialog(null, "Berhasil simpan ke excel ");
        File file = new File("D:/file_kelas_X.csv");
    }
}

```

```

        try {
            Desktop.getDesktop().open(file);
        } catch (IOException e) {
        }
    } catch (Exception e) {
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if ("Kelas X".equals(choice1.getSelectedItem().toString())) {
        System.out.println("Kelas X");
        cobatampilTabel(0);
    }
    if ("Kelas XI XII IPA".equals(choice1.getSelectedItem().toString())) {
        System.out.println("Kelas XI XII IPA");
        cobatampilTabelIPA(0);
    }
    if ("Kelas XI XII IPS".equals(choice1.getSelectedItem().toString())) {
        System.out.println("Kelas XI XII IPS");
        cobatampilTabelIPS(0);
    }
}

public void play() {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TabelJadwalOke().setVisible(true);
        }
    });
}

// public static void main(String args[]) {
//     TabelJadwalOke tabel = new TabelJadwalOke();
//     tabel.play();
// }

// Variables declaration - do not modify
private javax.swing.JButton ButtonExportExcel;
private java.awt.Choice choice1;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration

```

```

}
Jadwalvariabel.java
package ga.jadwal.code;

import ga.jadwal.gui.TabelJadwalOke;

/**
 * ALGORITMA GENETIKA UNTUK PENJADWALAN MAPEL SEKOLAH Pembentukan variabel
 * untuk
 * penjadwalan kelas Variabel ini masih STATIS, belum dinamis. variabel yg ada
 * disini: gen gen
 *
 * @author Andhika Lady Maharsi
 */
public class jadwal_variabel {
    // Variabel untuk keterangan jadwal //

    public static final int totalKelas = 4;
    public static final int[][] pertemuan = new int[jadwal_1_individu.SIZE][20][totalKelas];
    // 20 adalah jumlah pertemuan seminggu, 4 adalah jumlah kelas X
    public static final String[] nama_hari = new String[pertemuan[1].length];
    public static final int[] jumlah_pertemuan_seminggu = new int[80];
    public static final int[] jumlah_pertemuan_seminggu_random = new int[80];
    public static final int jumlahGuru = 20;
    public static final int[] jumlah_jam_guru = new int[jumlahGuru];
    public static final String[] nama_mapel = new String[jumlahGuru];
    public static final String[] nama_mapelIPA = new String[jumlahGuru];
    public static final String[] nama_mapelIPS = new String[jumlahGuru];

    public void inisialisasi() {
        //    buatIndividu();
        //    buatIndividuOlahraga();
        buatIndividuVertikal();
        setJumlahJam();
        setNamaMapel();
        setNamaMapelIPA();
        setNamaMapelIPS();
        setHari();
        //    LihatPertemuanVertikal();
        //    LihatPertemuanVertikalOke(19);
    }

    public void buatIndividu() { // nggak dipakai tapi penting
        int[] individu = new int[20];
        for (int k = 0; k < individu.length; k++) {
            individu[k] = individu.length + 1;
        }
        for (int i = 0; i < individu.length; i++) {

```

```

        int a = (int) (Math.random() * (individu.length));
        if (individu[a] == individu.length + 1) {
            individu[a] = i;
        } else {
            i = i - 1;
        }
    }
    for (int j = 0; j < individu.length; j++) {
        System.out.println(individu[j]);
    }
}

public void buatIndividuOlahraga() {
    for (int index = 0; indeks < jadwal_1_individu.SIZE; index++) {
        for (int ij = 0; ij < totalKelas; ij++) {
            int[] individuNot = new int[]{2, 3, 6, 7, 10, 11, 14, 15};
            int[] individu = new int[20];
            for (int k = 0; k < individu.length; k++) {
                individu[k] = individu.length + 1;
            }
            for (int l = 0; l < individuNot.length; l++) {
                individu[individuNot[l]] = individu.length + 2;
            }
            for (int i = 0; i < individu.length; i++) {
                int ab = 0;
                int tempat = (int) (Math.random() * (individu.length));
                if (individu[tempat] == individu.length + 1) {
                    individu[tempat] = ab;
                    i = individu.length;
                } else {
                    i = i + 1;
                }
            }
            for (int j = 0; j < individu.length; j++) {
                int a = (int) (Math.random() * (individu.length));
                if (individu[a] == individu.length + 1) {
                    individu[a] = j;
                } else {
                    j = j - 1;
                }
            }
            System.out.println("");
            for (int j = 0; j < individu.length; j++) {
                pertemuan[index][j][ij] = individu[j];
                System.out.println(individu[j]);
            }
        }
    }
}

```

```

    }
}

public void buatIndividuVertikal() {
    for (int indeks = 0; indeks < jadwal_1_individu.SIZE; index++) {
        for (int i = 0; i < totalKelas; i++) {
            int[] individu = new int[20];
            for (int k = 0; k < individu.length; k++) {
                individu[k] = individu.length + 1;
            }
            for (int j = 0; j < individu.length; j++) {
                int a = (int) (Math.random() * (individu.length));
                if (individu[a] == individu.length + 1) {
                    individu[a] = j;
                } else {
                    j = j - 1;
                }
            }
            for (int jk = 0; jk < individu.length; jk++) {
                // System.out.println(individu[jk]);
                pertemuan[index][jk][i] = individu[jk];
            }
            // System.out.println("");
        }
    }
}

public void setJumlahJam() {
    for (int i = 0; i < jumlahGuru; i++) {
        jumlah_jam_guru[i] = totalKelas;
    }
}

public void LihatPertemuanVertikal() {
    for (int index = 0; index < jadwal_1_individu.SIZE; index++) {
        System.out.println("");
        for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
            System.out.print(jadwal_variabel.nama_hari[i] + " --- > ");
            for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {
                System.out.print(jadwal_variabel.nama_mapel[jadwal_variabel.pertemuan[index][i][j]] +
                    ", ");
            }
            System.out.println("");
        }
        System.out.println("");
        for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
            System.out.print((index) + " --- > ");

```



```

        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {
            System.out.print(jadwal_variabel.pertemuan[index][i][j] + " , ");
        }
        System.out.println("");
    }
}

public void LihatPertemuanVertikalOke(int index) {
    System.out.println("\nHasil Akhir Jadwal Algoritma Genetika");
    // for (int i = 0; i < jadwal_variabel_kelas_X.pertemuan[index].length; i++) {
    //     System.out.print(jadwal_variabel_kelas_X.nama_hari[i]+" --- > ");
    //     for (int j = 0; j < jadwal_variabel_kelas_X.pertemuan[index][i].length; j++) {
    //         System.out.print(jadwal_variabel_kelas_X.nama_mapel[jadwal_variabel_kelas_X.pertemuan[index][i][j]] + " , ");
    //     }
    //     System.out.println("");
    // }
    System.out.println("");
    for (int i = 0; i < jadwal_variabel.pertemuan[index].length; i++) {
        System.out.print((index) + " --- > ");
        for (int j = 0; j < jadwal_variabel.pertemuan[index][i].length; j++) {
            System.out.print(jadwal_variabel.pertemuan[index][i][j] + " , ");
        }
        System.out.println("");
    }
}

public void setNamaMapel_kelasX() {
    nama_mapel[0] = "Olahraga";
    nama_mapel[1] = "Matematika 2";
    nama_mapel[2] = "Bahasa Indonesia 1";
    nama_mapel[3] = "Bahasa Indonesia 2";
    nama_mapel[4] = "Bahasa Inggris 1";
    nama_mapel[5] = "Bahasa Inggris 2";
    nama_mapel[6] = "Fisika";
    nama_mapel[7] = "Biologi";
    nama_mapel[8] = "Kimia";
    nama_mapel[9] = "Ekonomi";
    nama_mapel[10] = "Sosiologi";
    nama_mapel[11] = "Matematika 1";
    nama_mapel[12] = "TIK";
    nama_mapel[13] = "Kesenian";
    nama_mapel[14] = "Bahasa Jawa";
    nama_mapel[15] = "Bahasa Asing";
    nama_mapel[16] = "Bimbingan Konseling";
    nama_mapel[17] = "Sejarah";
}

```

```

        nama_mapel[18] = "Geografi";
        nama_mapel[19] = "Pendidikan Karakter";
    }

    public void setNamaMapel() {
        nama_mapel[0] = "Olahraga";
        nama_mapel[1] = "Fisika";
        nama_mapel[2] = "Biologi";
        nama_mapel[3] = "Kimia";
        nama_mapel[4] = "Bahasa Indonesia 1";
        nama_mapel[5] = "Bahasa Inggris";
        nama_mapel[6] = "Geografi";
        nama_mapel[7] = "Ekonomi";
        nama_mapel[8] = "Matematika 1";
        nama_mapel[9] = "Sejarah";
        nama_mapel[10] = "Sosiologi";
        nama_mapel[11] = "TIK";
        nama_mapel[12] = "Kesenian";
        nama_mapel[13] = "Bahasa Jawa";
        nama_mapel[14] = "Bahasa Asing";
        nama_mapel[15] = "BK";
        nama_mapel[16] = "Pendidikan Karakter";
        nama_mapel[17] = "Matematika 2";
        nama_mapel[18] = "Bahasa Indonesia 2";
        nama_mapel[19] = "Bahasa Inggris 2";
    }

    public void setNamaMapelIPA() {
        nama_mapelIPA[0] = "Olahraga";
        nama_mapelIPA[1] = "Matematika 2";
        nama_mapelIPA[2] = "Fisika 1";
        nama_mapelIPA[3] = "Fisika 2";
        nama_mapelIPA[4] = "Biologi 1";
        nama_mapelIPA[5] = "Biologi 2";
        nama_mapelIPA[6] = "Kimia 1";
        nama_mapelIPA[7] = "Kimia 2";
        nama_mapelIPA[8] = "Bahasa Indonesia 1";
        nama_mapelIPA[9] = "Bahasa Indonesia 2";
        nama_mapelIPA[10] = "Bahasa Inggris 1";
        nama_mapelIPA[11] = "Bahasa Inggris 2";
        nama_mapelIPA[12] = "Geografi";
        nama_mapelIPA[13] = "Sejarah";
        nama_mapelIPA[14] = "Matematika 1";
        nama_mapelIPA[15] = "TIK";
        nama_mapelIPA[16] = "Kesenian";
        nama_mapelIPA[17] = "Bahasa Jawa";
        nama_mapelIPA[18] = "Bahasa Asing";
        nama_mapelIPA[19] = "Bimbingan Konseling";
    }

```

```

}

public void setNamaMapelIPS() {
    nama_mapelIPS[0] = "Matematika 1";
    nama_mapelIPS[1] = "Matematika 2";
    nama_mapelIPS[2] = "Bahasa Indonesia 1";
    nama_mapelIPS[3] = "Bahasa Indonesia 2";
    nama_mapelIPS[4] = "Bahasa Inggris 1";
    nama_mapelIPS[5] = "Bahasa Inggris 2";
    nama_mapelIPS[6] = "Geografi 1";
    nama_mapelIPS[7] = "Geografi 2";
    nama_mapelIPS[8] = "Ekonomi 1";
    nama_mapelIPS[9] = "Ekonomi 2";
    nama_mapelIPS[10] = "Sosiologi 1";
    nama_mapelIPS[11] = "Sosiologi 2";
    nama_mapelIPS[12] = "Sejarah 1";
    nama_mapelIPS[13] = "Sejarah 2";
    nama_mapelIPS[14] = "Olahraga";
    nama_mapelIPS[15] = "TIK";
    nama_mapelIPS[16] = "Kesenian";
    nama_mapelIPS[17] = "Bahasa Jawa";
    nama_mapelIPS[18] = "Bahasa Asing";
    nama_mapelIPS[19] = "Bimbingan Konseling";
}

public void setHari() {
    nama_hari[0] = "Senin Jam 1-2";
    nama_hari[1] = "Jam 3-4";
    nama_hari[2] = "Jam 5-6";
    nama_hari[3] = "Jam 7-8";
    nama_hari[4] = "Selasa Jam 1-2";
    nama_hari[5] = "Jam 3-4";
    nama_hari[6] = "Jam 5-6";
    nama_hari[7] = "Jam 7-8";
    nama_hari[8] = "Rabu Jam 1-2";
    nama_hari[9] = "Jam 3-4";
    nama_hari[10] = "Jam 5-6";
    nama_hari[11] = "Jam 7-8";
    nama_hari[12] = "Kamis Jam 1-2";
    nama_hari[13] = "Jam 3-4";
    nama_hari[14] = "Jam 5-6";
    nama_hari[15] = "Jam 7-8";
    nama_hari[16] = "Jumat Jam 1-2";
    nama_hari[17] = "Jam 3-4";
    nama_hari[18] = "Sabtu Jam 1-2";
    nama_hari[19] = "Jam 3-4";
}

```

```

public void tukarDataGen(int index1, int index2) {
    try {
        for (int i = 0; i < jadwal_variabel.pertemuan[index1].length; i++) {
            System.arraycopy(jadwal_variabel.pertemuan[index2][i], 0,
                jadwal_variabel.pertemuan[index1][i], 0, jadwal_variabel.pertemuan[index1][i].length);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
    }
}

// public static void main(String[] args) {
//     jadwal_variabel jadvar = new jadwal_variabel();
//     TabelJadwalOke tab = new TabelJadwalOke();
//     jadvar.inisialisasi();
//     tab.play();
// }
}

```

Jadwal1individu.java

```

package ga.jadwal.code;

import ga.jadwal.gui.Input;
import ga.jadwal.gui.TabelJadwalOke;
import javax.swing.JOptionPane;

/**
 * ALGORITMA GENETIKA UNTUK PENJADWALAN MAPEL SEKOLAH 1. Inisialisasi Individu
 * dan pengolahan nilai fitness variabel yg terletak disini = nilai error,
 * fitness, probabilitas
 *
 * @author Andhika Lady Maharsi
 */
public class jadwal_1_individu {

    public static final int SIZE = 20; // jumlah individu yang akan dibangkitkan per generasi
    public static final int[] jadwal = new int[SIZE];
    public static final double[] nilai_error = new double[SIZE];
    public static final double[] fitness = new double[SIZE];
    public static final double[] probabilitas = new double[SIZE];
    static final int[] nilaiCrossOver = new int[SIZE];
    static final int[] nilaiMutasi = new int[SIZE];
    static double total_fitnes;
    private boolean ulang = true;
    private static int loop = 0;
    private static int iterasi;
    private int crossoverOK = 1;
    private int mutasiOK = 1;
    static int total_probabilitas;
    private static int mainLoop;

```

```

// public static final int mainLoop = iterasi;// nilai ini diubah-ubah sesuai jumlah
generasi yang diinginkan.
public static final double[] elitisme = new double[mainLoop];
public static final double[] nilai_error_e = new double[mainLoop];
public static final double[] fitness_e = new double[mainLoop];
public static final double[] probabilitas_e = new double[mainLoop];
jadwal_variabel jadvar = new jadwal_variabel();
jadwal_3_crossOver_new cross = new jadwal_3_crossOver_new();
jadwal_4_mutasi mutasi = new jadwal_4_mutasi();
TabelJadwalOke tabelOke = new TabelJadwalOke();
// input inp = new input();
// private int mainloop;

public void setNilaiError(int index, double gene) {
    nilai_error[index] = gene;
}

public void setFitness(int index, double gene) {
    fitness[index] = gene;
}

public void setProbabilitas(int index, double gene) {
    probabilitas[index] = gene;
}

public void setLoop(int index){
    mainLoop = index;
}

public void goToCross2() { // ke crossover sempurna
    System.out.println("");
    for (int i = 0; i < jadwal_1_individu.SIZE - 2; i++) {
        nilaiCrossOver[i] = (int) (Math.random() * 1000);
        cross.crossOver(nilaiCrossOver[i], i, (i + 2), i, i + 2);
    }
}

public void goToMutasi() {
    System.out.println("");
    for (int i = 0; i < SIZE; i++) {
        nilaiMutasi[i] = (int) (Math.random() * 1000);
        mutasi.mutasi(nilaiMutasi[i], i);
    }
}

public void inisialisasi() {
    System.out.println("Iterasi = " + iterasi);
    System.out.println("Mainloop = " + mainLoop);
}

```

```

jadvar.inisialisasi();
evaluasiIndividu();
tampilAwal(SIZE);

while (loop < mainLoop) {
    System.out.print("halo");System.out.print(loop); System.out.print("halo");
    jadwal_2_SeleksiRoulette Roulette = new jadwal_2_SeleksiRoulette();
    Roulette.inisialisasi();
    if(tampilAwal(SIZE) == true) {
        //loop = mainLoop;
        break;
    }
    goToCross2();
    if(tampilAwal(SIZE) == true) {
        //loop = mainLoop;
        break;
    }
    goToMutasi();
    if(tampilAwal(SIZE) == true) {
        //loop = mainLoop;
        break;
    }
    sortingNilaiError(loop);
    evaluasiIndividu();
    loop++;
}

if (loop == mainLoop) {
    JOptionPane.showMessageDialog(null, "Eksekusi Algoritma Genetika Penjadwalan
belum mencapai fitness maksimal\nUlangi eksekusi atau naikan Maksimal Generasi");
}
else{
    tabelOke.play();
}
}

public void evaluasiIndividu() {
    evaluasiNilaiError(SIZE);
    evaluasiFitness(SIZE);
    evaluasiProbabilitas();
}

public void evaluasiNilaiError(int indexIndividu) { // constraint baru tabrakan
    int li = 0;
    while (li < indexIndividu) {
        int k, m;
        nilai_error[li] = 0;
        for (int i = 0; i < jadwal_variabel.pertemuan[li].length; i++) {

```

```

        for (int j = 0; j < jadwal_variabel.pertemuan[li][i].length; j++) {    // Nilai error
untuk tabrakan per jam
            k = jadwal_variabel.pertemuan[li][i][j];
            for (int l = j + 1; l < jadwal_variabel.pertemuan[li][i].length; l++) {
                m = jadwal_variabel.pertemuan[li][i][l];
                if (k == m) {
                    nilai_error[li] = nilai_error[li] + 1;
                }
            }
        }
    }
    for (int i = 0; i < jadwal_variabel.pertemuan[li].length; i++) {    // Nilai error untuk
tabrakan per kolom
        for (int l = i + 1; l < jadwal_variabel.pertemuan[li].length; l++) {
            for (int j = 0; j < jadwal_variabel.pertemuan[li][i].length; j++) {    // Nilai error
untuk tabrakan per kelas
                k = jadwal_variabel.pertemuan[li][i][j];
                m = jadwal_variabel.pertemuan[li][l][j];
                if (k == m) {
                    nilai_error[li] = nilai_error[li] + 1;
                }
            }
        }
    }
    for (int i = 0; i < jadwal_variabel.pertemuan[li].length; i++) {
        for (int l = i + 1; l < jadwal_variabel.pertemuan[li].length; l++) {
            for (int j = 0; j < jadwal_variabel.pertemuan[li][i].length; j++) {
                if (jadwal_variabel.pertemuan[li][2][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][3][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][6][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][7][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][10][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][11][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
                if (jadwal_variabel.pertemuan[li][14][j] == 0){
                    nilai_error[li] = nilai_error[li] + 1;
                }
            }
        }
    }

```

```

        if (jadwal_variabel.pertemuan[li][15][j] == 0){
            nilai_error[li] = nilai_error[li] + 1;
        }
    }
}

setNilaiError(li, nilai_error[li]);
li++;
}
}

public void evaluasiFitness(int indexIndividu) { // atur nilai fitness
    int li = 0;
    while (li < indexIndividu) {
        double nilaiError = nilai_error[li];
        double fit = 1 / nilaiError;
        double fitkuadrat = fit * 1000;
        setFitness(li, fitkuadrat);
        li++;
    }
}

public double evaluasiProbabilitas() { // atur nilai probabilitas
    double total_eror = 0.0;
    double total_fitness = 0.0;
    for (int i = 0; i < jadwal.length; i++) {
        total_eror += nilai_error[i];
        total_fitness += fitness[i];
    }
    total_fitnes = total_fitness;
    System.out.println("Total nilai error = " + total_eror);
    System.out.println("Total nilai fitness = " + total_fitness);
    for (int i = 0; i < jadwal.length; i++) {
        double prob;
        prob = (fitness[i] / total_fitnes) * 1000;
        setProbabilitas(i, prob);
    }
    double total_proba = 0.0;
    for (int i = 0; i < jadwal.length; i++) {
        total_proba += (int) probabilitas[i];
    }
    total_probabilitas = (int) total_proba;
    System.out.println("Total nilai probabilitas = " + total_probabilitas);
    return total_eror;
}

public boolean tampilAwal(int indek) { // untuk menampilkan semuanya

```



```

System.out.println("");
System.out.println("Menampilkan nilai fitness per individu");
int l = 0;
while (l < indek) {
    System.out.println("Nilai error individu " + l + " adalah = " + nilai_error[l] + " Fitness
= " + fitness[l] + " Probabilitas = " + probabilitas[l]);
    if (nilai_error[l] == 0) { // jika hasil generate sudah memenuhi syarat jadwal
yang benar.
        tukarData(0, l);
//        jadvar.tukarDataGen(0, l);
        jadvar.LihatPertemuanVertikalOke(0);
        jadvar.LihatPertemuanVertikalOke(l);
//        jadvar.LihatPertemuanVertikal();

        l = indek+1;
        loop = mainLoop+1;
        return true;
    }
    l++;
}
return false;
}

public void sortingNilaiError(int indexElit) { // selection sort
    for (int i = 0; i < jadwal_1_individu.SIZE - 1; i++) {
        int temp = i;
        for (int j = i + 1; j < jadwal_1_individu.SIZE; j++) {
            if (jadwal_1_individu.nilai_error[j] < jadwal_1_individu.nilai_error[temp]) {
                temp = j;
            }
            if (temp != i) {
                double tempsementara = jadwal_1_individu.nilai_error[i];
                tukarData(i, temp);
                //jadwal_1_individu_updated_perkelas.nilai_error[i] =
jadwal_1_individu_updated_perkelas.nilai_error[temp];
                jadwal_1_individu.nilai_error[temp] = tempsementara;
            }
        }
    }
    Setelitism(indexElit, jadwal_1_individu.nilai_error[0], jadwal_1_individu.fitness[0],
jadwal_1_individu.probabilitas[0]);
}

public void tukarData(int index1, int index2) {
    jadwal_1_individu.nilai_error[index1] = jadwal_1_individu.nilai_error[index2];
    jadwal_1_individu.fitness[index1] = jadwal_1_individu.fitness[index2];
    jadwal_1_individu.probabilitas[index1] = jadwal_1_individu.probabilitas[index2];
    jadvar.tukarDataGen(index1, index2);
}

```

```

    }

    public void Setelitism(int index, double n, double f, double p) {
        try {
            jadwal_1_individu.nilai_error_e[index] = n;
            jadwal_1_individu.fitness_e[index] = f;
            jadwal_1_individu.probabilitas_e[index] = p;
        } catch (ArrayIndexOutOfBoundsException e) {
        }
    }

    public void sortingElitisme() {
        for (int i = 0; i < jadwal_1_individu.mainLoop - 1; i++) {
            int temp = i;
            for (int j = i + 1; j < jadwal_1_individu.mainLoop; j++) {
                if (jadwal_1_individu.nilai_error_e[j] < jadwal_1_individu.nilai_error_e[temp]) {
                    temp = j;
                }
                if (temp != i) {
                    double tempsementara = jadwal_1_individu.nilai_error_e[i];
                    tukarDataElit(i, temp);
                    //jadwal_1_individu_updated_perkelas.nilai_error[i] =
                    jadwal_1_individu_updated_perkelas.nilai_error[temp];
                    jadwal_1_individu.nilai_error_e[temp] = tempsementara;
                }
            }
        }
    }

    public void tukarDataElit(int index1, int index2) {
        jadwal_1_individu.nilai_error_e[index1] = jadwal_1_individu.nilai_error_e[index2];
        jadwal_1_individu.fitness_e[index1] = jadwal_1_individu.fitness_e[index2];
        jadwal_1_individu.probabilitas_e[index1] =
        jadwal_1_individu.probabilitas_e[index2];
        jadvar.tukarDataGen(index1, index2);
    }

    public static void main(String[] args) {
        jadwal_1_individu jadwal1 = new jadwal_1_individu();
        jadwal_variabel jadvar = new jadwal_variabel();
        jadwal_2_SeleksiRoulette jadwal2 = new jadwal_2_SeleksiRoulette();
        jadwal_3_crossOver_new jadwal3 = new jadwal_3_crossOver_new();
        jadwal_4_mutasi jadwal4 = new jadwal_4_mutasi();
        TabelJadwalOke tab = new TabelJadwalOke();
        jadwal1.inisialisasi();
    }
}

```

Jadwal_2_seleksi.java

```

package ga.jadwal.code;

/**
 * variabel yg ada disini: index probabilitas,index hasil roulette
 * @author Andhika Lady Maharsi
 */
public class jadwal_2_SeleksiRoulette {

    jadwal_variabel jadvar = new jadwal_variabel();
    jadwal_1_individu jadwal = new jadwal_1_individu();
    jadwal_3_crossOver_new cross = new jadwal_3_crossOver_new();
    static final int[] nilaiCrossOver = new int[jadwal_1_individu.SIZE];
    jadwal_4_mutasi jadwal4 = new jadwal_4_mutasi();
    static final int[] indexProbabilitas = new int[jadwal_1_individu.SIZE];
    static final int[] arrayRoulette = new int[1000];
    static final int[] hasilRoulette = new int[jadwal_1_individu.SIZE];

    public void inisialisasi() {
        setIndexProbabilitas();
//        tampil();
        setArrayUntukRoulette();
        RouletteWheel();
        jadwal.evalusiIndividu();
    }

    public void setIndexProbabilitas() { //method untuk menentukan index
        int i = 0;
        while (i < jadwal_1_individu.SIZE) {
            if (i == 0) {
                indexProbabilitas[i] = (int) jadwal_1_individu.probabilitas[i];
            } else {
                indexProbabilitas[i] = (int) (indexProbabilitas[i - 1] +
jadwal_1_individu.probabilitas[i]);
            }
            i++;
        }
    }

    public void setArrayUntukRoulette() { //method untuk menentukan array probabilitas
        untuk masing2 individu
        int i = 0;
        while (i < jadwal_1_individu.SIZE) {
            if (i == (jadwal_1_individu.SIZE - 1)) {
                for (int j = (int) indexProbabilitas[jadwal_1_individu.SIZE - 2]; j <
jadwal_1_individu.total_probabilitas; j++) {
                    arrayRoulette[j] = i;
                }
            } else if (i == 0) {

```

```

        for (int j = 0; j < indexProbabilitas[0]; j++) {
            arrayRoulette[j] = i;
        }
    } else {
        for (int j = (int) indexProbabilitas[i - 1]; j < indexProbabilitas[i]; j++) {
            arrayRoulette[j] = i;
        }
    }
    i++;
}
}

public void RouletteWheel() {    // method utama seleksi Roulette
    System.out.println("");
    System.out.println("Hasil seleksi roulette");
    int i = 0;
    while (i < jadwal_1_individu.SIZE) {
        int getRandom = (int) (Math.random() * 1000);
        int pilih = arrayRoulette[getRandom];
        //System.out.println("Hasil random = " + getRandom + " Individu yang terpilih
adalah " + pilih);
        hasilRoulette[i] = pilih;
//        tampilHasilRoulette(pilih);
        jadvar.tukarDataGen(i, pilih);
        i++;
    }
}

public void tampilHasilRoulette(int l) { //untuk menampilkan hasil roulette
    System.out.println("Nilai error individu " + l + " adalah = " +
jadwal_1_individu.nilai_error[l] + " Fitness = " + jadwal_1_individu.fitness[l] + "
Probabilitas = " + jadwal_1_individu.probabilitas[l]);
}

public void tampilHasilRouletteSetelahGantiGen() {
    System.out.println("");
    System.out.println("Hasil Individu setelah ganti gen");
    for (int l = 0; l < jadwal_1_individu.SIZE; l++) {
        System.out.println("Nilai error individu " + l + " adalah = " +
jadwal_1_individu.nilai_error[l] + " Fitness = " + jadwal_1_individu.fitness[l] + "
Probabilitas = " + jadwal_1_individu.probabilitas[l]);
    }
}

public void tampil() {
    System.out.println("");
    for (int i = 0; i < jadwal_1_individu.SIZE; i++) {
        System.out.println("Index probabilitas individu " + i + " = " + indexProbabilitas[i]);
    }
}

```

```

    }
}

// public static void main(String[] args) {
//     jadwal_2_SeleksiRoulette jadvar = new jadwal_2_SeleksiRoulette();
//     jadvar.inisialisasi();
// }
}

Jadwal_3_crossover.java
package ga.jadwal.code;

/**
 * crossover sempurna
 * @author Andhika Lady Maharsi
 */
public class jadwal_3_crossOver_new {

    static final int crossoverRate = 300;
    jadwal_variabel jadvar = new jadwal_variabel();

    public void crossOver(int nilaiCross, int index, int index2, int pilih, int pilih2) {
        int k = (int) (Math.random()*jadwal_variabel.totalKelas);
        int k1 = jadwal_variabel.totalKelas - k;

        // System.out.println("Hasil random "+k+" dan pasangannya "+k1);

        if (nilaiCross < crossoverRate) { // jika kurang dari crossover rate, maka individu
            dikawinsilang
            // System.out.println("Sebelum di Cross over = .....");
            // tampilCrossOver(pilih);
            //
            // System.out.println("One cut point dari indeks ke = "+k);
            for (int i = 0; i < k; i++) {
                for (int j = 0; j < 20; j++) {
                    jadwal_variabel.pertemuan[pilih2][j][i] =
                    jadwal_variabel.pertemuan[pilih][j][i];
                }
            }
            for (int i = k; i < jadwal_variabel.totalKelas ; i++) {
                for (int j = 0; j < 20; j++) {
                    jadwal_variabel.pertemuan[pilih][j][i] =
                    jadwal_variabel.pertemuan[pilih2][j][i];
                }
            }
            // System.out.println("Setelah di Cross over = .....");
            // tampilCrossOver(pilih);
        }
        else { // individu tidak dikawin silang karena tidak memenuhi syarat crossover

```

```

        for (int i = k; i < jadwal_variabel.pertemuan[pilih].length ; i++) {
            System.arraycopy(jadwal_variabel.pertemuan[pilih][i], 0,
            jadwal_variabel.pertemuan[pilih][i], 0, jadwal_variabel.pertemuan[pilih][i].length);
        }
    }
}

public void tampilCrossOver(int pilih) {
    for (int i = 0; i < jadwal_variabel.pertemuan[pilih].length; i++) {
        System.out.print((i+1)+" --- > ");
        for (int j = 0; j < jadwal_variabel.pertemuan[pilih][i].length; j++) {
            System.out.print(jadwal_variabel.pertemuan[pilih][i][j] + " , ");
        }
        System.out.println("");
    }
    System.out.println("");
}
}

```

Jadwal_4_mutasi.java

package ga.jadwal.code;

```

/**
 *
 * @author Andhika Lady Maharsi
 */
public class jadwal_4_mutasi {

    static final int[] mutasiRate = new int[jadwal_1_individu.SIZE];
    static final int indeksmutasi = 100;
    jadwal_3_crossOver_new jadwal3 = new jadwal_3_crossOver_new();

    public void mutasi(int il, int pilih) {
        mutasiRate[pilih] = il;
        //System.out.println("Index "+i+" mutasi rate: "+j);
        if (il < indeksmutasi) {
            int kl = (int) (Math.random() * jadwal_variabel.totalKelas);
            int[] individu = new int[20];
            for (int k = 0; k < individu.length; k++) {
                individu[k] = individu.length + 1;
            }
            for (int i = 0; i < individu.length; i++) {
                int a = (int) (Math.random() * (individu.length));
                if (individu[a] == individu.length + 1) {
                    individu[a] = i;
                } else {
                    i = i - 1;
                }
            }
        }
    }
}

```

```

        for (int j = 0; j < individu.length; j++) {
//            System.out.println(individu[j]);
            jadwal_variabel.pertemuan[pilih][j][kl] = individu[j];
        }
        //jadwal3.tampilCrossOver(pilih);
    }
}

public void gantiIndividu(int index, int pilih) {
    for (int i = 0; i < jadwal_variabel.pertemuan[pilih].length; i++) {
        System.arraycopy(jadwal_variabel.pertemuan[pilih][i], 0,
jadwal_variabel.pertemuan[index][i], 0, jadwal_variabel.pertemuan[pilih][i].length);
    }
}

// public static void main(String[] args) {
//     jadwal_4_mutasi jadvar = new jadwal_4_mutasi();
//     jadvar.inisialisasi();
// }
}

```

Lampiran Tabel 15. Kode Program

Lampiran 6. Salinan Validitas

Komentar :

Saran :

- Perbaiki interface yang mudah dipahami guru
- Perbaiki parameter mapel olahraga.

Kesimpulan :

Instrumen *Usability* dan *Functionally* Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika ini dinyatakan :

- a. Layak dilakukan uji coba ke lapangan tanpa revisi.
- b. Layak dilakukan uji coba ke lapangan dengan revisi sesuai saran.
- c. Belum layak dilakukan uji coba ke lapangan.

Ahli Materi

Kuswari Hermawati, M. Kom

Lampiran Gambar 6. Salinan Validitas Ahli 1.

SUB INDIKATOR FAKTOR USABILITY ISO

No	Butir Pertanyaan	Sub Indikator			
		U	L	O	A
1	Apakah kegunaan program dapat dikenali dari awal interface?				
2	Apakah bahasa yang ditampilkan mudah dipahami?				
3	Apakah hasil output tabel penjadwalan mudah dipahami?				
4	Apakah pesan error yang ada mudah dipahami?				
5	Apakah warna huruf sudah sesuai?				
6	Apakah ukuran huruf sudah sesuai?				
7	Apakah jenis huruf sudah sesuai?				
8	Apakah letak tombol pada program sudah sesuai?				
9	Apakah ukuran jendela program sudah sesuai?				
10	Apakah ukuran tombol sudah sesuai?				
11	Apakah keterangan pada tombol informatif dilihat? mudah dipahami				
12	Apakah tata letak menu yang ada informatif dilihat? mudah dipahami				
13	Apakah tabel penjadwalan informatif dilihat? mudah dipahami				
14	Apakah menu-menu yang ada mudah diingat kembali?				
15	Apakah menu-menu yang ada dapat mudah diakses?				
16	Apakah Anda dapat mengubah tampilan paket kelas tanpa kesulitan?				
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?				
18	Apakah tombol-tombol yang ada mudah digunakan?				
19	Apakah menu bantuan yang ada cukup membantu?				
20	Apakah combobox penggantian kelas mudah digunakan?				
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?				

U : Understandability

L : Learnability

O : Operability

A : Attractiveness

Revisi :

.....

.....

.....

.....

.....



Komentar :

.....

.....

.....

.....

.....

Saran :

- Tambahkan warning untuk input bernilai negatif.
- Tambahkan keterangan pada bagian test case.
- Penyimpanan file excel dapat langsung ditampilkan

.....

.....

.....

Kesimpulan :

Instrumen *Usability* dan *Functionally* Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika ini dinyatakan :

- a. Layak dilakukan uji coba ke lapangan tanpa revisi.
- ☒ b. Layak dilakukan uji coba ke lapangan dengan revisi sesuai saran.
- c. Belum layak dilakukan uji coba ke lapangan.

Ahli Materi

Ahli *Pratiyo*

Lampiran Gambar 7. Salinan Validitas Ahli 2.

Lampiran 7. Hasil Angket

**USABILITY TEST SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH
MENGUNAKAN ALGORITMA GENETIKA**

Nama : Ariel Budiman

Jabatan : Guru TIK

Berilah tanda cek (v) pada pilihan jawaban poin kesetujuan sesuai persepsi masing-masing untuk penelitian "Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika" dengan poin sebagai berikut :

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

No	Butir Pertanyaan	Poin Kesetujuan				
		1	2	3	4	5
1	Apakah kegunaan program dapat dikenali dari awal interface?		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
2	Apakah bahasa yang ditampilkan mudah dipahami?		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
3	Apakah hasil output tabel penjadwalan mudah dipahami?				<input checked="" type="checkbox"/>	
4	Apakah pesan error yang ada mudah dipahami?			<input checked="" type="checkbox"/>		
5	Apakah kekontrasan warna huruf sudah sesuai?				<input checked="" type="checkbox"/>	
6	Apakah ukuran huruf sudah sesuai?				<input checked="" type="checkbox"/>	
7	Apakah jenis huruf sudah sesuai?				<input checked="" type="checkbox"/>	
8	Apakah letak tombol pada program sudah sesuai?				<input checked="" type="checkbox"/>	
9	Apakah ukuran jendela program sudah sesuai?				<input checked="" type="checkbox"/>	
10	Apakah ukuran tombol sudah sesuai?				<input checked="" type="checkbox"/>	
11	Apakah keterangan pada tombol mudah dipahami?				<input checked="" type="checkbox"/>	
12	Apakah tata letak menu yang ada mudah dipahami?				<input checked="" type="checkbox"/>	
13	Apakah tabel penjadwalan mudah dipahami?				<input checked="" type="checkbox"/>	
14	Apakah menu-menu yang ada mudah diingat kembali?					<input checked="" type="checkbox"/>

15	Apakah menu-menu yang ada dapat mudah diakses?					✓
16	Apakah Anda dapat memilih tampilan paket kelas tanpa kesulitan?					✓
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?					✓
18	Apakah tombol-tombol yang ada mudah digunakan?				✓	
19	Apakah menu bantuan yang ada cukup membantu?				✓	
20	Apakah combobox penggantian kelas mudah digunakan?				✓	
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?					✓

Saran :

Perlu menu input data mata pelajaran, guru dan guru mengajar mapel.
 mengakomodir kebutuhan terknik: seperti hari, bulan, jam, dan ada jadwal guru yg terintegrasi juga tidak ada kempas dan jam pagi dll.

Mengetahui,


 Arief Budiman

Lampiran Gambar 8. Salinan Hasil Angket Pengguna 1.

**USABILITY TEST SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH
MENGUNAKAN ALGORITMA GENETIKA**

Nama : Imam Pujiadi
Jabatan : Guru / Kurikulum

Berilah tanda cek (V) pada pilihan jawaban poin kesetujuan sesuai persepsi masing-masing untuk penelitian "Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika" dengan poin sebagai berikut :

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

No	Butir Pertanyaan	Poin Kesetujuan				
		1	2	3	4	5
1	Apakah kegunaan program dapat dikenali dari awal interface?				✓	
2	Apakah bahasa yang ditampilkan mudah dipahami?				✓	
3	Apakah hasil output tabel penjadwalan mudah dipahami?					✓
4	Apakah pesan error yang ada mudah dipahami?				✓	
5	Apakah kontras warna huruf sudah sesuai?					✓
6	Apakah ukuran huruf sudah sesuai?					✓
7	Apakah jenis huruf sudah sesuai?					✓
8	Apakah letak tombol pada program sudah sesuai?				✓	
9	Apakah ukuran jendela program sudah sesuai?				✓	
10	Apakah ukuran tombol sudah sesuai?					✓
11	Apakah keterangan pada tombol mudah dipahami?					✓
12	Apakah tata letak menu yang ada mudah dipahami?					✓
13	Apakah tabel penjadwalan mudah dipahami?				✓	
14	Apakah menu-menu yang ada mudah diingat kembali?				✓	

15	Apakah menu-menu yang ada dapat mudah diakses?				✓	
16	Apakah Anda dapat memilih tampilan paket kelas tanpa kesulitan?					✓
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?					✓
18	Apakah tombol-tombol yang ada mudah digunakan?				✓	
19	Apakah menu bantuan yang ada cukup membantu?					✓
20	Apakah combobox penggantian kelas mudah digunakan?					✓
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?					✓

Saran :

- Ditambahkan input untuk mapel
- Customisasi jumlah kelas
- Customisasi keterangan guru

Mengetahui,

[Signature]
Iman Puspadi

Lampiran Gambar 9. Salinan Hasil Angket Pengguna 2.

USABILITY TEST SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGUNAKAN ALGORITMA GENETIKA

Nama : Nanik Supriyati
Jabatan : Guru / kurikulum

Berilah tanda cek (v) pada pilihan jawaban poin kesetujuan sesuai persepsi masing-masing untuk penelitian "Sistem Penjadwalan Mata Pelajaran Sekolah menggunakan Algoritma Genetika" dengan poin sebagai berikut :

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

No	Butir Pertanyaan	Poin Kesetujuan				
		1	2	3	4	5
1	Apakah kegunaan program dapat dikenali dari awal interface?				✓	
2	Apakah bahasa yang ditampilkan mudah dipahami?				✓	
3	Apakah hasil output tabel penjadwalan mudah dipahami?					✓
4	Apakah pesan error yang ada mudah dipahami?				✓	
5	Apakah kontras warna huruf sudah sesuai?					✓
6	Apakah ukuran huruf sudah sesuai?					✓
7	Apakah jenis huruf sudah sesuai?					✓
8	Apakah letak tombol pada program sudah sesuai?					✓
9	Apakah ukuran jendela program sudah sesuai?					✓
10	Apakah ukuran tombol sudah sesuai?				✓	
11	Apakah keterangan pada tombol mudah dipahami?				✓	
12	Apakah tata letak menu yang ada mudah dipahami?					✓
13	Apakah tabel penjadwalan mudah dipahami?				✓	
14	Apakah menu-menu yang ada mudah diingat kembali?				✓	



15	Apakah menu-menu yang ada dapat mudah diakses?				✓	
16	Apakah Anda dapat memilih tampilan paket kelas tanpa kesulitan?					✓
17	Apakah Anda dapat mengexport hasil penjadwalan dalam bentuk excel tanpa kesulitan?				✓	✓
18	Apakah tombol-tombol yang ada mudah digunakan?					✓
19	Apakah menu bantuan yang ada cukup membantu?				✓	
20	Apakah combobox penggantian kelas mudah digunakan?					✓
21	Apakah tombol export xls hasil penjadwalan mudah digunakan?					✓

Saran :

.....

.....

.....

.....

.....


Mengetahui,



Nanik Supriyati

Lampiran Gambar 10. Salinan Hasil Angket Pengguna 3.

Lampiran 8. Surat Menyurat



PEMERINTAH KABUPATEN SLEMAN
BADAN PERENCANAAN PEMBANGUNAN DAERAH
 Jalan Parasamiya Nomor 1 Beran, Tridadi, Sleman, Yogyakarta 55511
 Telepon (0274) 868800, Faksimilie (0274) 868800
 Website: slemankab.go.id, E-mail : bappeda@slemankab.go.id

SURAT IZIN
 Nomor : 070 / Bappeda / 1953 / 2013
TENTANG
PENELITIAN
KEPALA BADAN PERENCANAAN PEMBANGUNAN DAERAH

Dasar : Keputusan Bupati Sleman Nomor : 55/Kep.KDH/A/2003 tentang Izin Kuliah Kerja Nyata, Praktek Kerja Lapangan, dan Penelitian.
 Menunjuk : Surat dari Sekretariat Daerah Pemerintah Daerah Daerah Istimewa Yogyakarta
 Nomor : 070/4565/V/5/2013 Tanggal : 28 Mei 2013
 Hal : Izin Penelitian

MENGIZINKAN :

Kepada :
 Nama : ANDHIKA LADY MAHARSI
 No.Mhs/NIM/NIP/NIK : 08520241041
 Program/Tingkat : SI
 Instansi/Perguruan Tinggi : Universitas Negeri Yogyakarta
 Alamat instansi/Perguruan Tinggi : Karangmalang, Yogyakarta 55281
 Alamat Rumah : Jl. Deresan 2 No. 18 Deresan, Condongcatur, Depok, Sleman, Yk.
 No. Telp / HP : 085227555610
 Untuk : Mengadakan Penelitian / Pra Survey / Uji Validitas / PKL dengan judul
SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA
 Lokasi : SMA Negeri 1 Kalasan, Sleman
 Waktu : Selama 3 bulan mulai tanggal: 28 Mei 2013 s/d 28 Agustus 2013


Dengan ketentuan sebagai berikut :

1. Wajib melapor diri kepada Pejabat Pemerintah setempat (Camat/ Kepala Desa) atau Kepala Instansi untuk mendapat petunjuk seperlunya.
2. Wajib menjaga tata tertib dan mentaati ketentuan-ketentuan setempat yang berlaku.
3. Izin tidak disalahgunakan untuk kepentingan-kepentingan di luar yang direkomendasikan.
4. Wajib menyampaikan laporan hasil penelitian berupa 1 (satu) CD format PDF kepada Bupati diserahkan melalui Kepala Badan Perencanaan Pembangunan Daerah.
5. Izin ini dapat dibatalkan sewaktu-waktu apabila tidak dipenuhi ketentuan-ketentuan di atas.

Demikian ijin ini dikeluarkan untuk digunakan sebagaimana mestinya, diharapkan pejabat pemerintah/non pemerintah setempat memberikan bantuan seperlunya.
 Setelah selesai pelaksanaan penelitian Saudara wajib menyampaikan laporan kepada kami 1 (satu) bulan setelah berakhirnya penelitian.

Dikeluarkan di Sleman
 Pada Tanggal : 30 Mei 2013
 a.n. Kepala Badan Perencanaan Pembangunan Daerah

Sekretaris
 u.b.
 Kepala Bidang Pengendalian dan Evaluasi



[Signature]
Dra. SUCITRI ANI SINURAYA, M.Si, M.M
 Pembina, IVa
 NIP. 19630112 198903 2 003

Tembusan :

1. Bupati Sleman (sebagai laporan)
2. Kepala Kantor Kesatuan Bangsa Kab. Sleman
3. Kepala Dinas Dikpora Kab. Sleman
4. Kabid. Sosial Budaya Bappeda Kab. Sleman
5. Camat Kalasan
6. Kepala SMA Negeri 1 Kalasan, Sleman
7. Wakil Dekan I Fak. Teknik UNY.
8. Yang Bersangkutan

Lampiran Gambar 11. Salinan Surat Ijin Penelitian Bappeda.

27/05/2013 7:38:00



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI YOGYAKARTA
FAKULTAS TEKNIK

Alamat : Kampus Karangmalang, Yogyakarta, 55281
Telp. (0274) 586168 psw. 276,289,292 (0274) 586734 Fax. (0274) 586734
website : <http://ft.uny.ac.id> e-mail: ft@uny.ac.id : teknik@uny.ac.id



Certificate No. QSC 00592

Nomor : 1704/UN34.15/PL/2013
Lamp. : 1 (satu) bendel
Hal : Permohonan Ijin Penelitian

27 Mei 2013

Yth.

1. Gubernur Provinsi DIY c.q. Ka. Biro Administrasi Pembangunan Setda Provinsi DIY
2. Bupati Sleman c.q. Kepala Bappeda Kabupaten Sleman
3. Kepala Dinas Pendidikan, Pemuda dan Olahraga Propinsi DIY
4. Kepala Dinas Pendidikan Kabupaten Sleman
6. Kepala / Direktur/ Pimpinan : SMA N 1 Kalasan

Dalam rangka pelaksanaan Skripsi kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan penelitian dengan judul **"SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA"**, bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta tersebut di bawah ini:

No.	Nama	NIM	Jurusan/Prodi	Lokasi Penelitian
	Andhika Lady Maharsi	08520241041	Pend. Teknik Informatika - S1	SMA N 1 KALASAN

Dosen Pembimbing/Dosen Pengampu : Dr. Eko Marpanaji
NIP : 19670608 199303 1 001

Adapun pelaksanaan penelitian dilakukan mulai tanggal 27 Mei 2013 sampai dengan selesai.

Demikian permohonan ini, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.



Dekan,
Wakil Dekan I,

Dr. Sunaryo Soenarto
NIP 19580630 198601 1 001

Tembusan:
Ketua Jurusan

08520241041 No. 1240



PEMERINTAH DAERAH DAERAH ISTIMEWA YOGYAKARTA
SEKRETARIAT DAERAH

Kompleks Kepatihan, Danurejan, Telepon (0274) 562811 - 562814 (Hunting)
YOGYAKARTA 55213

SURAT KETERANGAN / IJIN

070/4565/V/5/2013

Membaca Surat : Wakil Dekan I Fak. Teknik UNY Nomor : 1704/ UN34.15/PL/2013
Tanggal : 27 Mei 2013 Perihal : Ijin Penelitian

Mengingat : 1. Peraturan Pemerintah Nomor 41 Tahun 2006, tentang Perizinan bagi Perguruan Tinggi Asing, Lembaga Penelitian dan Pengembangan Asing, Badan Usaha Asing dan Orang Asing dalam melakukan Kegiatan Penelitian dan Pengembangan di Indonesia;

2. Peraturan Menteri Dalam Negeri Nomor 33 Tahun 2007, tentang Pedoman penyelenggaraan Penelitian dan Pengembangan di Lingkungan Departemen Dalam Negeri dan Pemerintah Daerah;

3. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 37 Tahun 2008, tentang Rincian Tugas dan Fungsi Satuan Organisasi di Lingkungan Sekretariat Daerah dan Sekretariat Dewan Perwakilan Rakyat Daerah.

4. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 18 Tahun 2009 tentang Pedoman Pelayanan Perizinan, Rekomendasi Pelaksanaan Survei, Penelitian, Pendataan, Pengembangan, Pengkajian, dan Studi Lapangan di Daerah Istimewa Yogyakarta.

DIJINKAN untuk melakukan kegiatan survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan kepada:

Nama : ANDHIKA LADY MAHARSI NIP/NIM : 08520241041
 Alamat : KARANGMALANG, YOGYAKARTA
 Judul : SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA
 Lokasi : SMA N 1 KALASAN Kota/Kab. SLEMAN
 Waktu : 28 Mei 2013 s/d 28 Agustus 2013

Dengan Ketentuan

1. Menyerahkan surat keterangan/ijin survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan *) dari Pemerintah Daerah DIY kepada Bupati/Walikota melalui institusi yang berwenang mengeluarkan ijin dimaksud;
2. Menyerahkan soft copy hasil penelitiannya baik kepada Gubernur Daerah Istimewa Yogyakarta melalui Biro Administrasi Pembangunan Selda DIY dalam compact disk (CD) maupun mengunggah (upload) melalui website adbang.jogjaprov.go.id dan menunjukkan cetakan asli yang sudah disahkan dan dibubuhi cap institusi;
3. Ijin ini hanya dipergunakan untuk keperluan ilmiah, dan pemegang ijin wajib mentaati ketentuan yang berlaku di lokasi kegiatan;
4. Ijin penelitian dapat diperpanjang maksimal 2 (dua) kali dengan menunjukkan surat ini kembali sebelum berakhir waktunya setelah mengajukan perpanjangan melalui website adbang.jogjaprov.go.id;
5. Ijin yang diberikan dapat dibatalkan sewaktu-waktu apabila pemegang ijin ini tidak memenuhi ketentuan yang berlaku.

Dikeluarkan di Yogyakarta

Pada tanggal 28 Mei 2013

A.n Sekretaris Daerah

Asisten Perencanaan dan Pembangunan

Replika Biro Administrasi Pembangunan

Hendar Susilowati, SH

NIP. 19580120 198503 2 003

Tembusan :

1. Yth. Gubernur Daerah Istimewa Yogyakarta (sebagai laporan);
2. Bupati Sleman, cq Bappeda
3. Ka. Dinas Pendidikan Pemuda dan Olahraga DIY
4. Wakil Dekan I Fak. Teknik UNY
5. Yang Bersangkutan

Lampiran Gambar 13. Salinan Surat Ijin Penelitian Gubernur.



PEMERINTAH KABUPATEN SLEMAN
DINAS PENDIDIKAN, PEMUDA DAN OLAHRAGA

SMA NEGERI 1 KALASAN

Alamat : Bogem, Tamanmartani, Kalasan, Sleman 55571
Telp. (0274) 496040, Fax.: (0274)496040
Email : sman1kalasan.sleman@gmail.com
Website : www.sman1kalasan.sch.id

SURAT KETERANGAN

Nomor : 07.0/526

Yang bertanda tangan dibawah ini Kepala Sekolah Menengah Atas Negeri 1 Kalasan Kabupaten Sleman menerangkan dengan sesungguhnya bahwa Mahasiswa Universitas Negeri Yogyakarta (UNY) tersebut di bawah ini :

N a m a	: ANDHIKA LADY MAHARSI
N I M	: 08520241041
Fakultas	: Teknik
Prodi	: Teknik Elektronika
Jurusan	: Pendidikan Teknik Informatika
Program/Tingkat	: Sarjana (S1)
Alamat Rumah	: Jl. Deresan II No.18 Deresan Sleman Yogyakarta

Telah melaksanakan kegiatan penelitian dengan judul "SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH MENGGUNAKAN ALGORITMA GENETIKA" di SMA Negeri 1 Kalasan pada tanggal 29 Juli 2013.

Demikian surat keterangan ini kami buat, untuk digunakan sebagaimana mestinya.

Kalasan, 29 Juli 2013
Kepala Sekolah,

Dis. H. TRI SUGIHARTO
NIP. 19570707 198103 1 024

Lampiran Gambar 14. Salinan Surat Keterangan Selesai Penelitian.